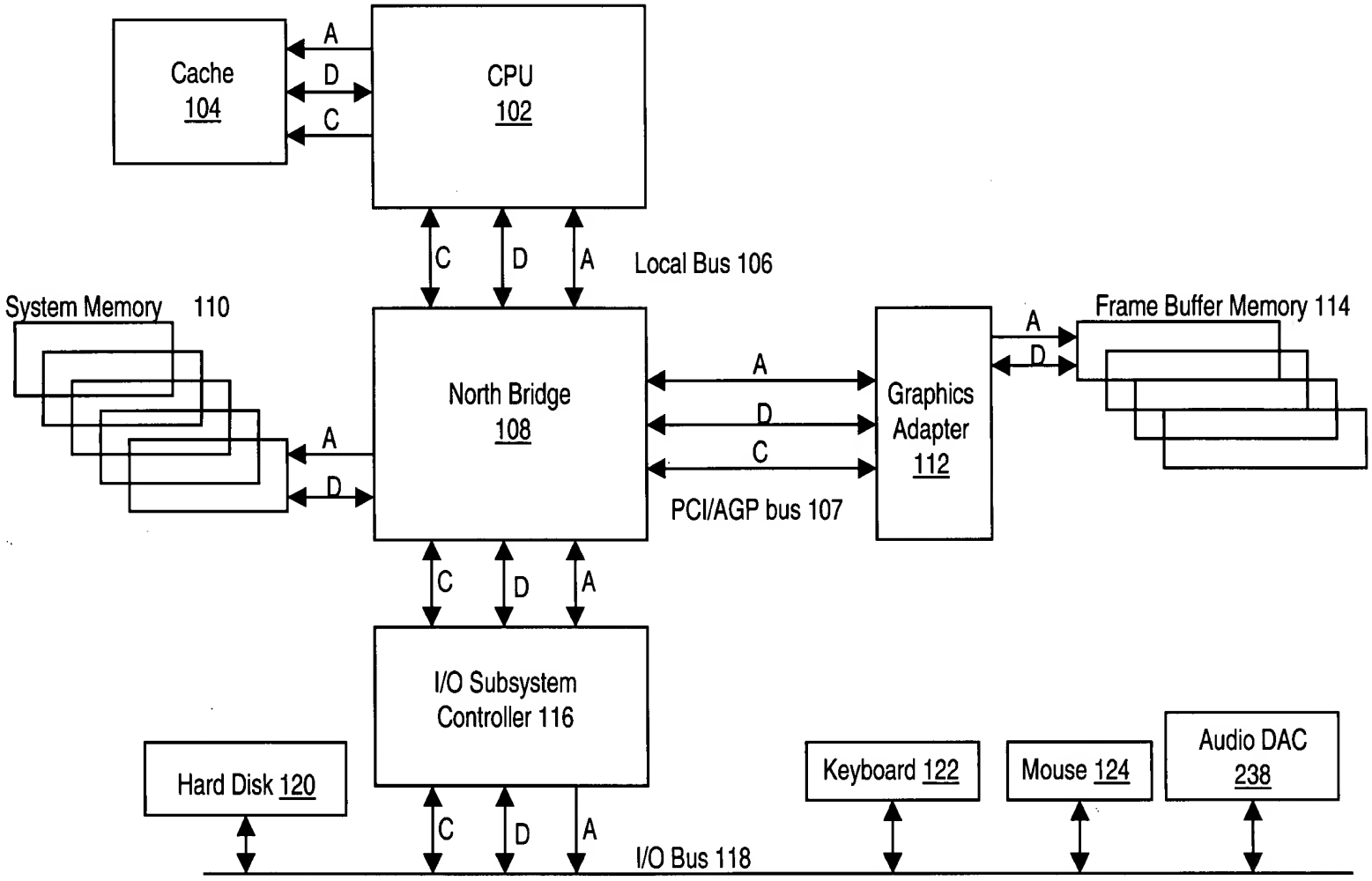


Figure 1
(Prior Art)

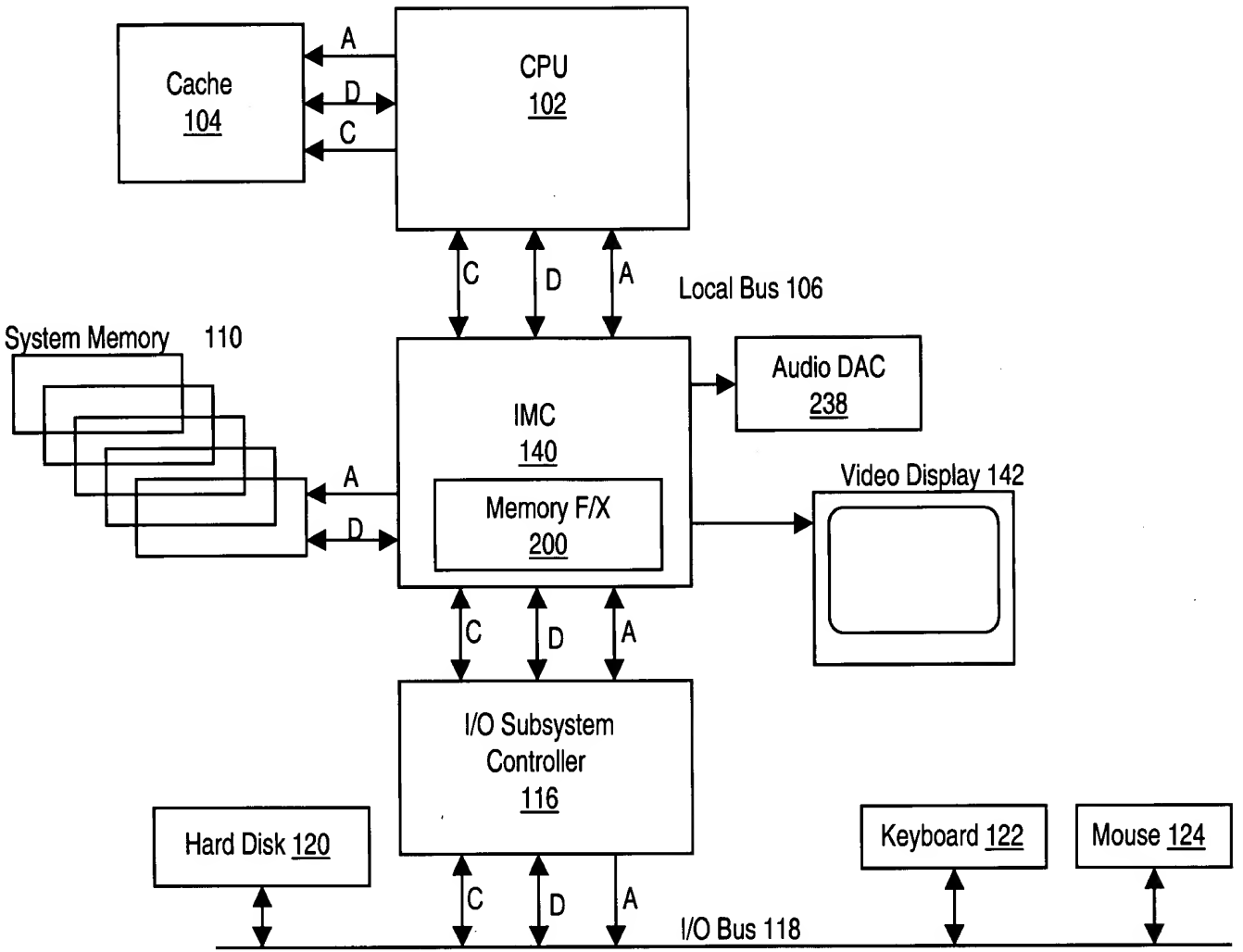

Cache
104
A
D
C
CPU
102
C
D
A
Local Bus 106
System Memory 110
IMC
140
Memory F/X
200
Audio DAC
238
Video Display 142
I/O Subsystem
Controller
116
Hard Disk 120
Keyboard 122
Mouse 124
I/O Bus 118

Figure 2a

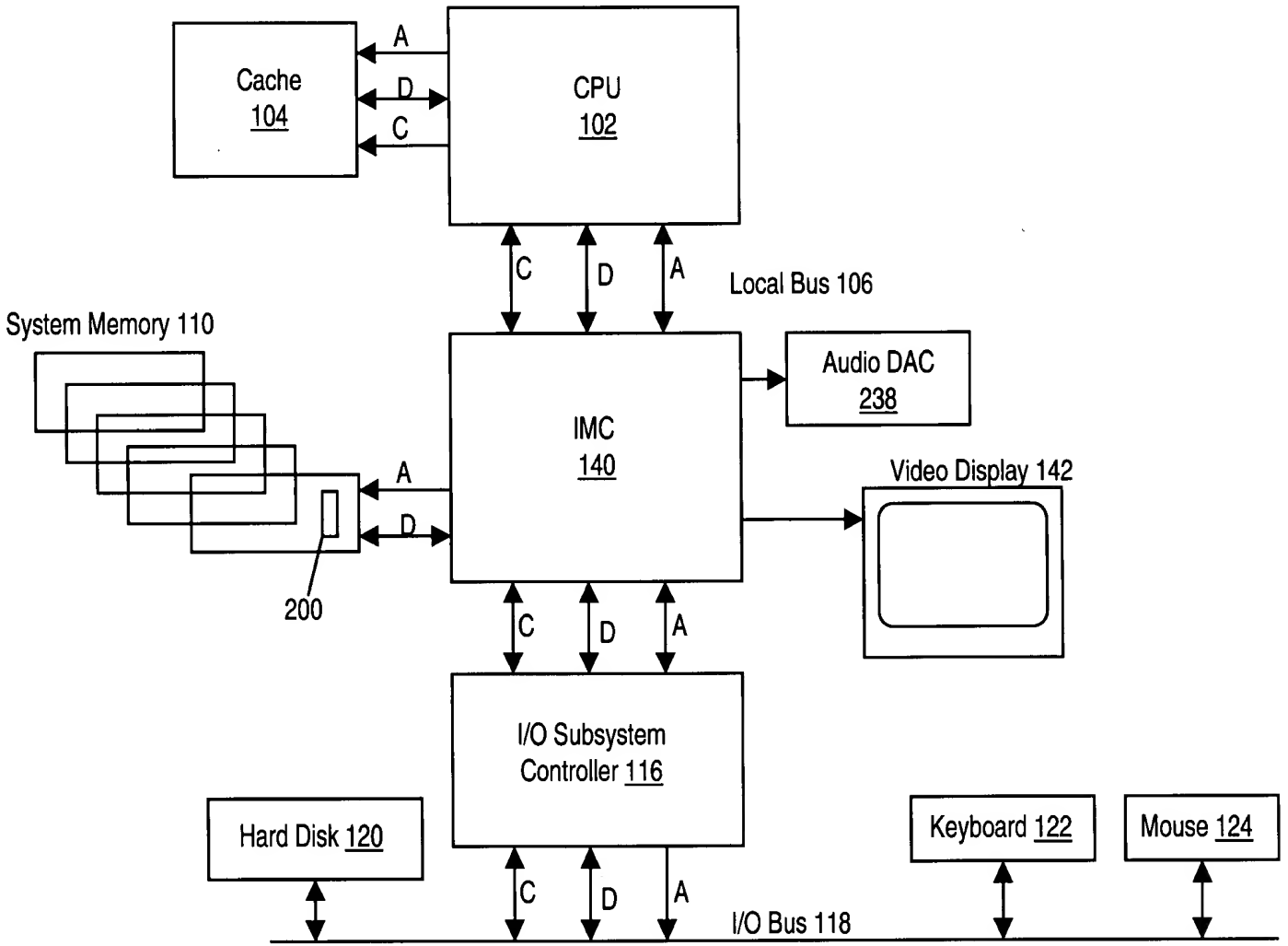


Figure 2b

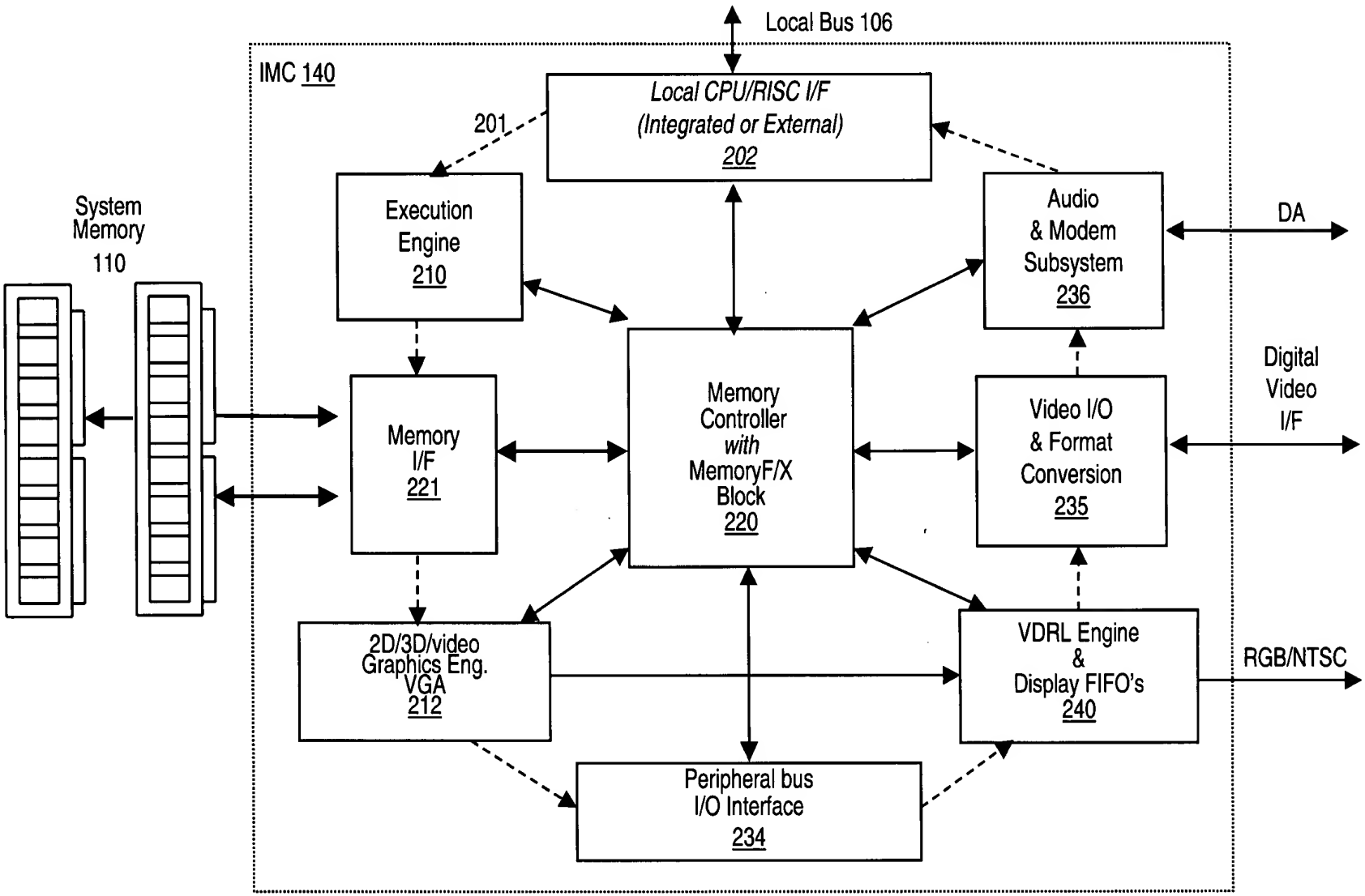


Figure 3

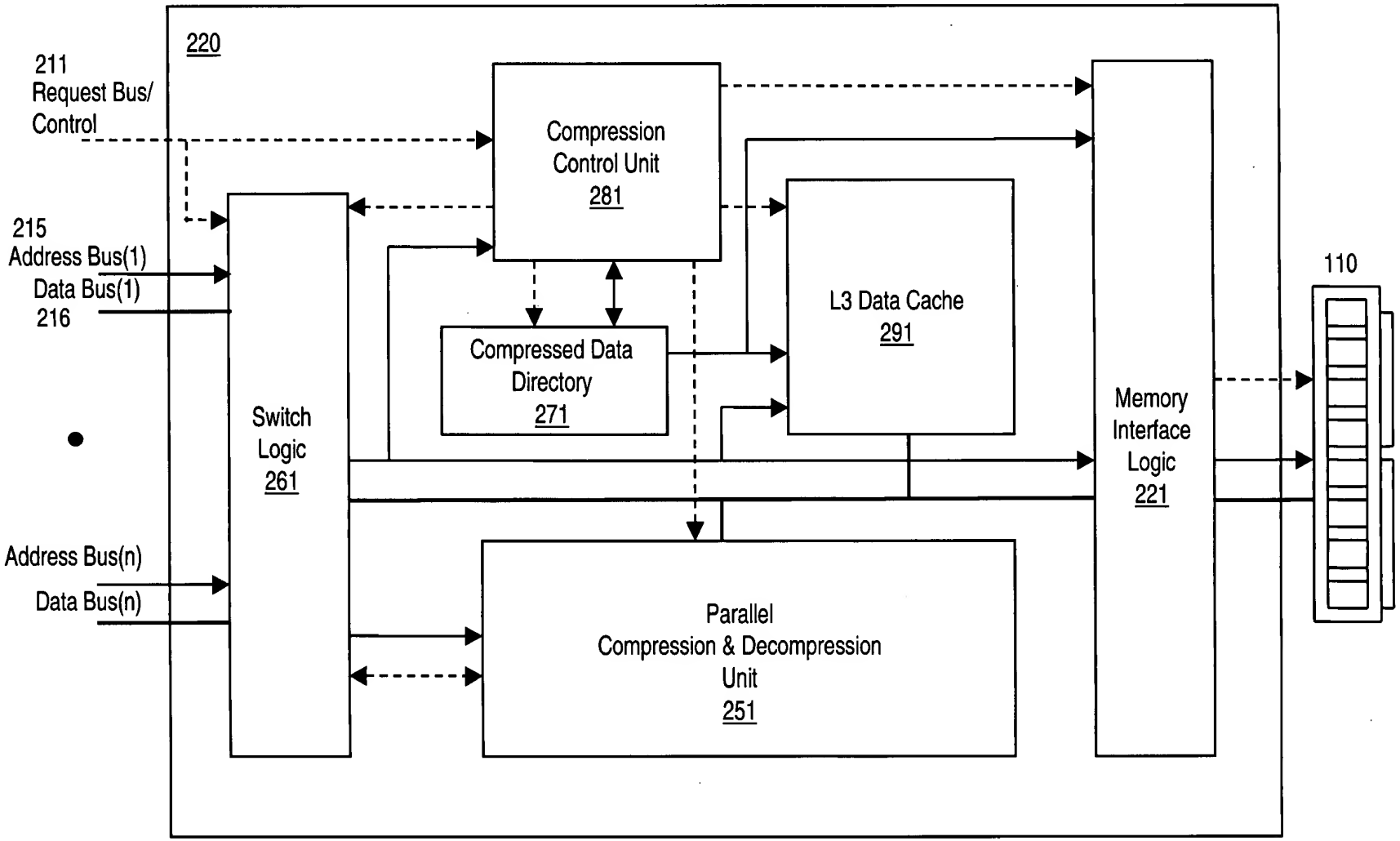
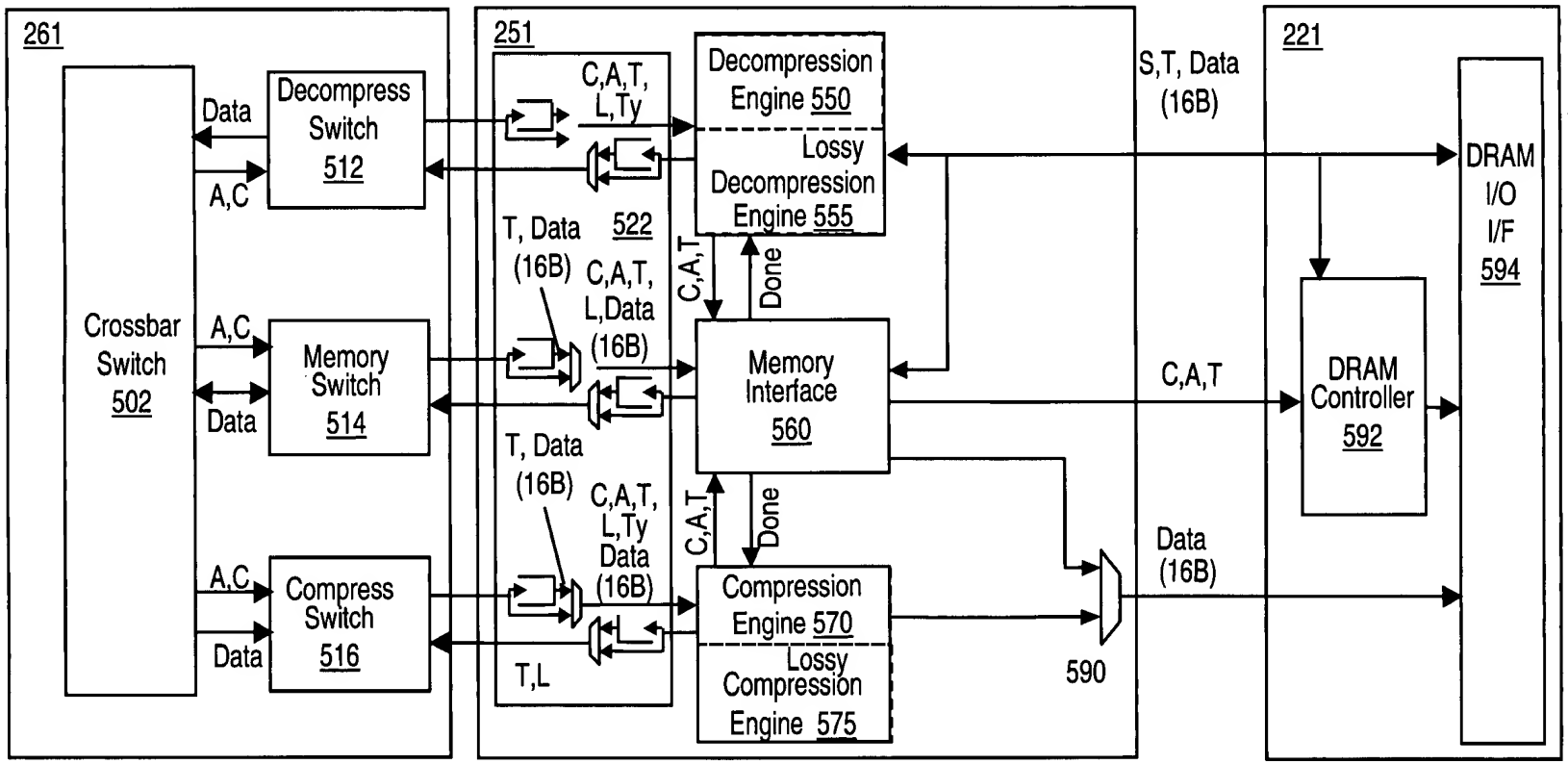


Figure 4



Key:
C – Command
A – Address
T – Tag
L – Length
Ty – Type
S – Status

Figure 5

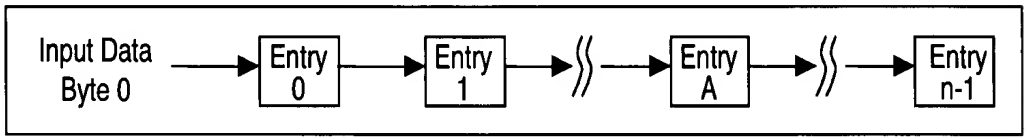


Figure 6a
(Prior Art)

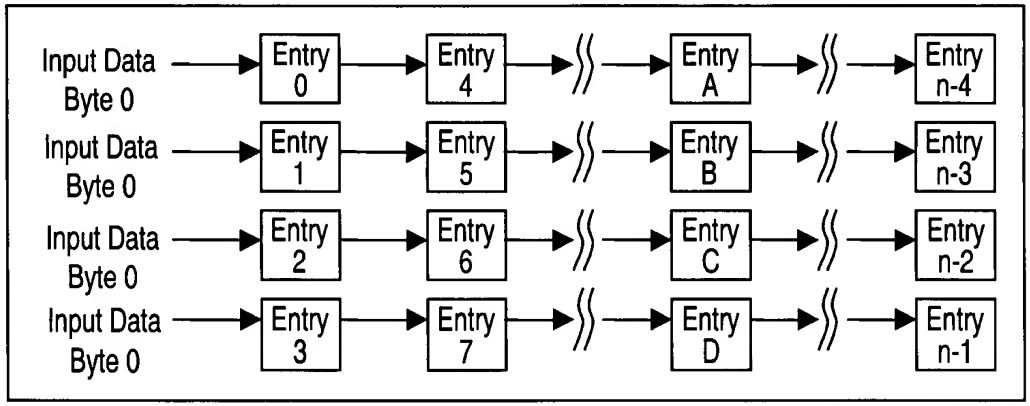


Figure 6b
(New Art)

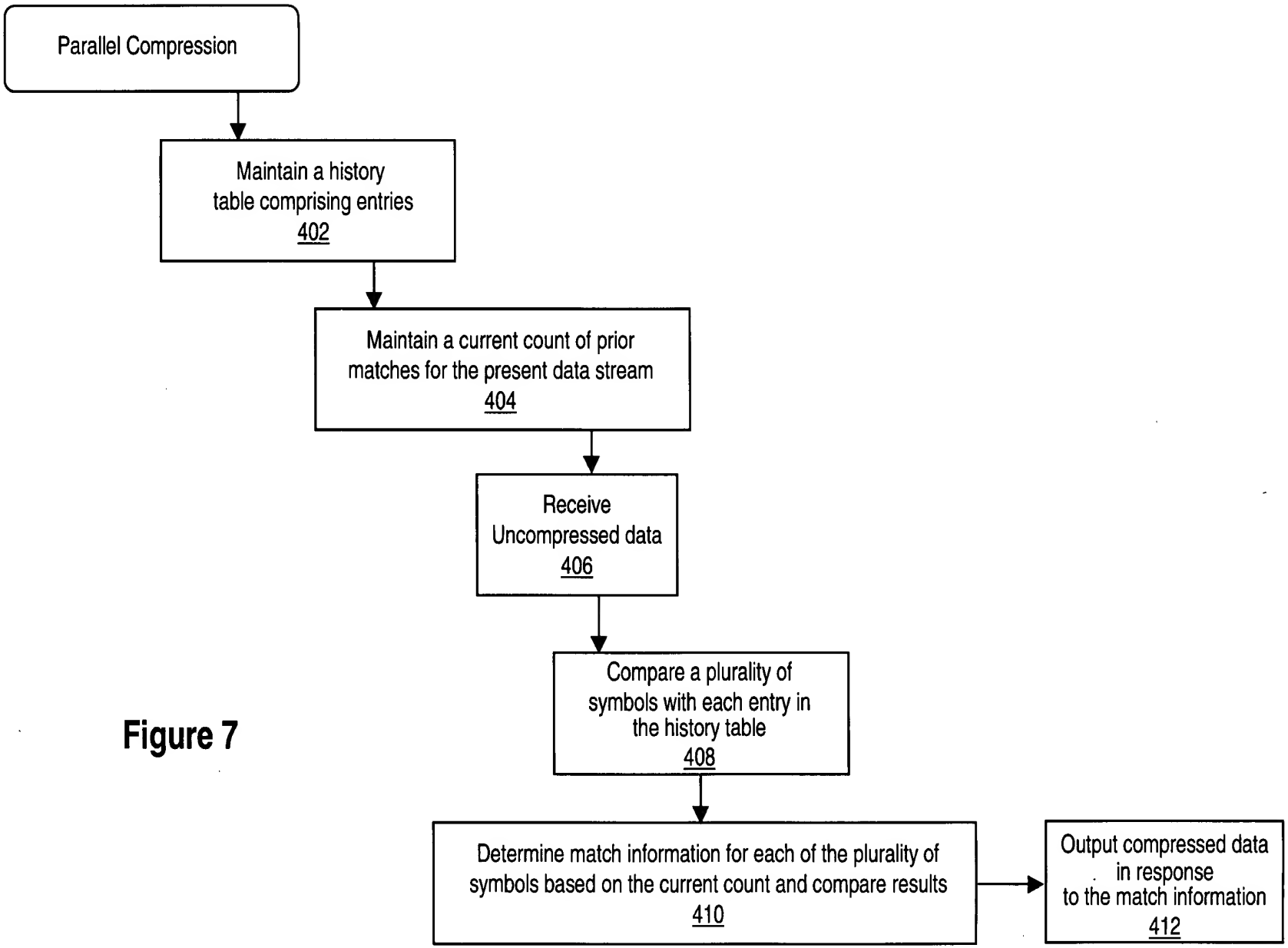


Figure 7

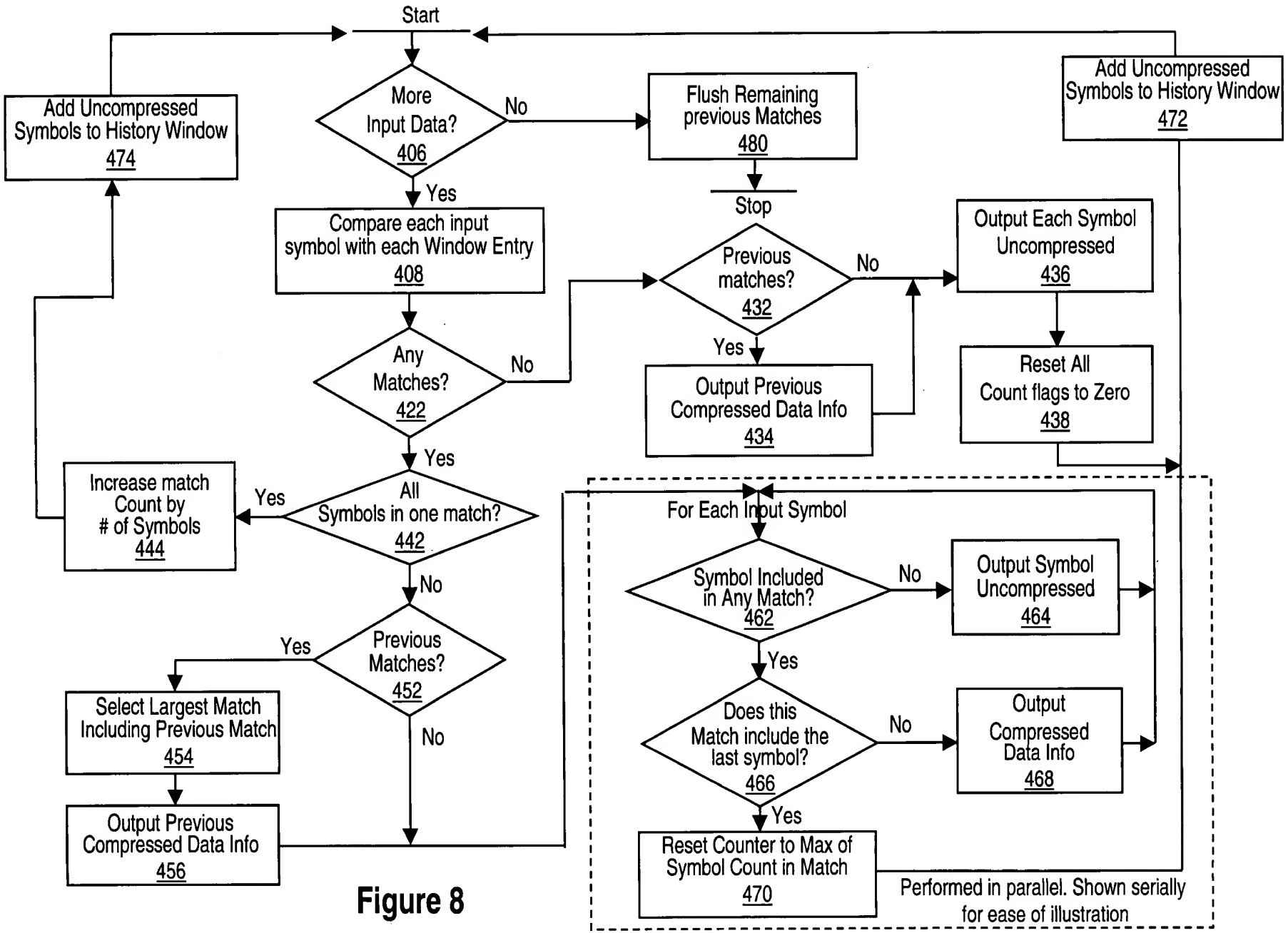


Figure 8

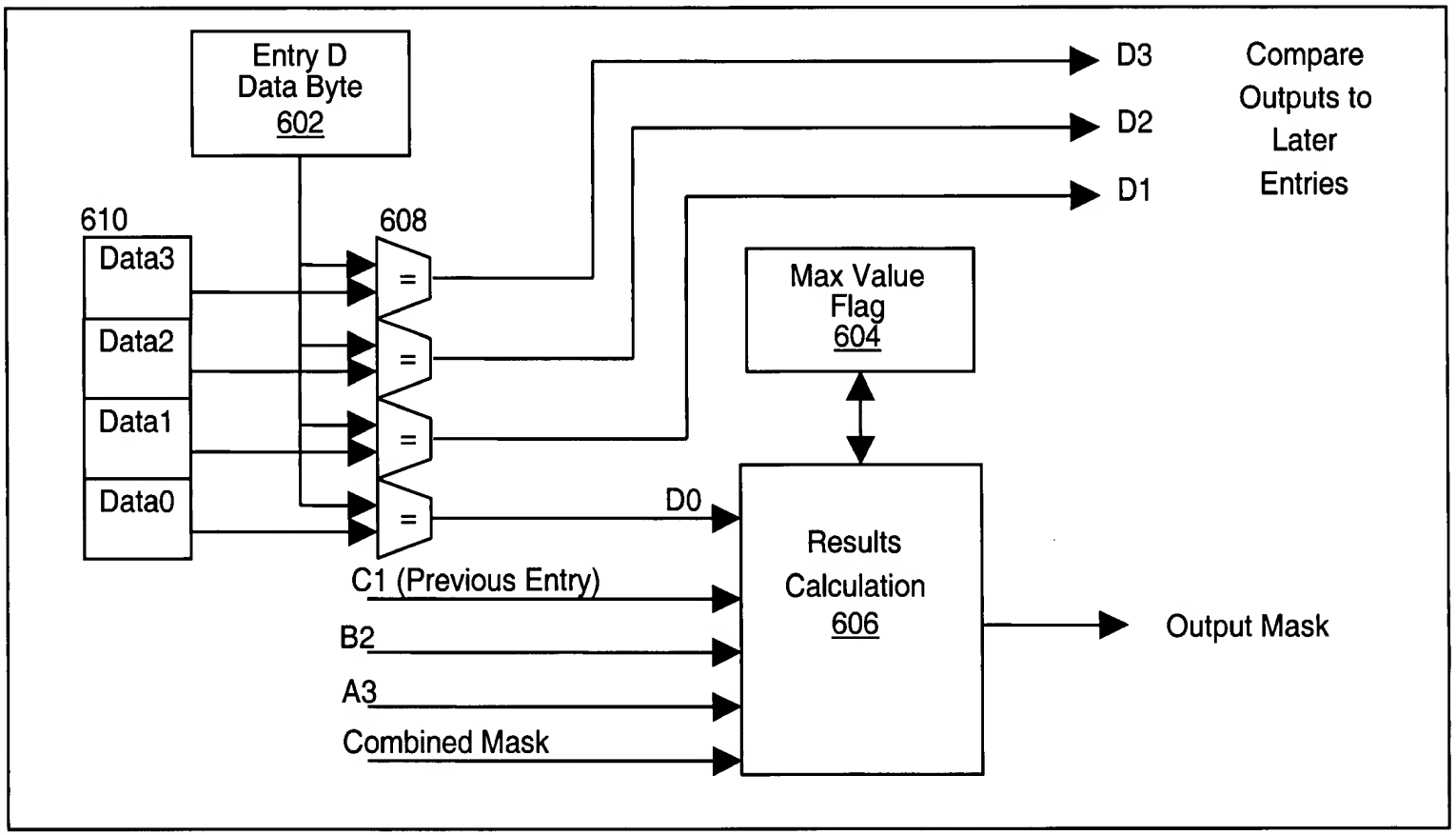


Figure 9

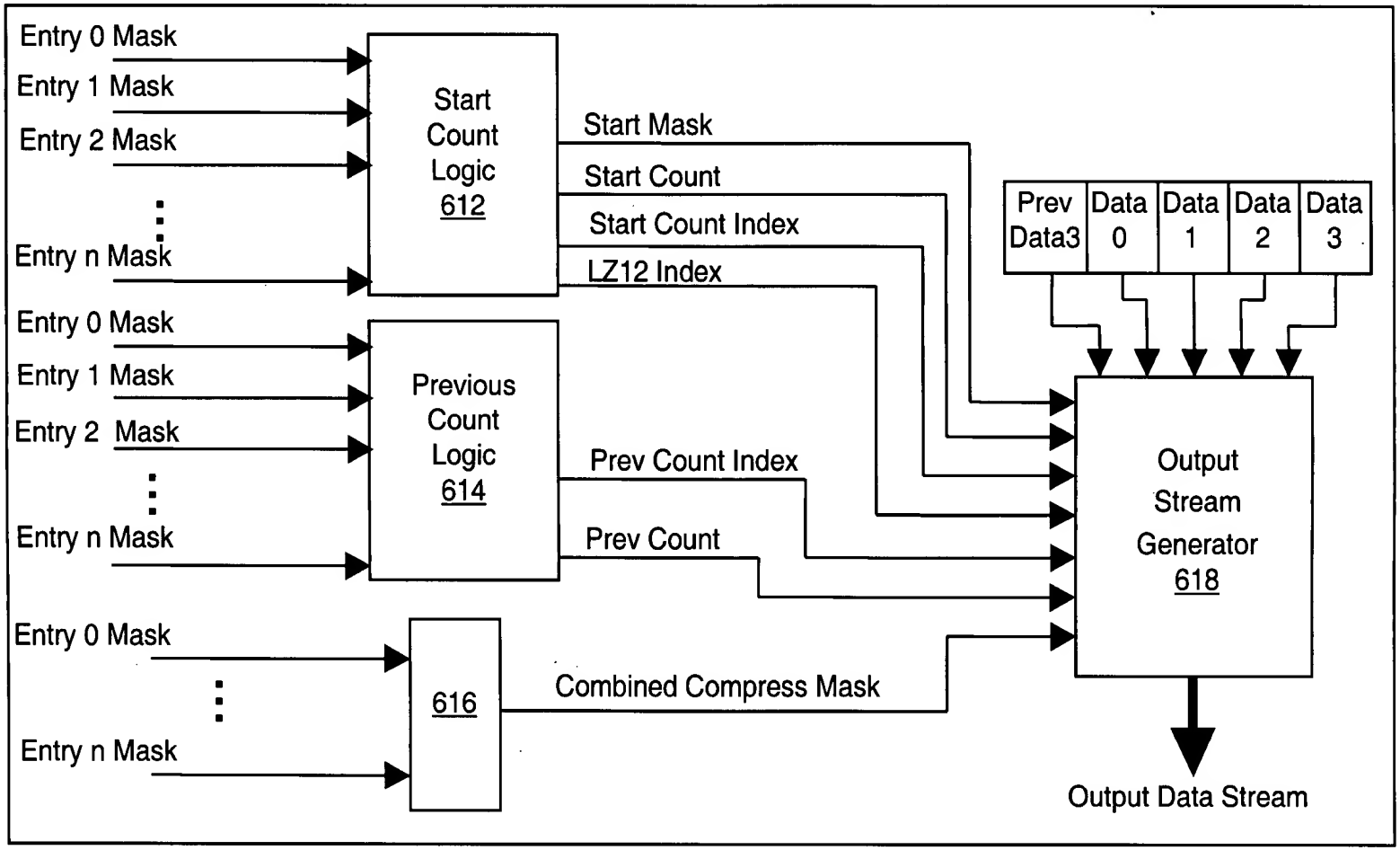


Figure 10

Input Matches				New Counter Value	Output Counter	Output Mask	Reset Value
D0	C1	B2	A3				
1	1	1	1	Saved+4	Saved +4	10000	0
1	1	1	0	0	Saved+3	10001	1
1	1	0	1	1	Saved+2	10010	2
1	1	0	0	0	Saved+2	10011	2
1	0	1	1	2	Saved+1	10100	3
1	0	1	0	0	Saved+1	10101	3
1	0	0	1	1	Saved+1	10110	3
1	0	0	0	0	Saved+1	10111	3
0	1	1	1	3	Saved	11000	4
0	1	1	0	0	Saved	01111	1
0	1	0	1	1	Saved	11010	4
0	1	0	0	0	Saved	11011	4
0	0	1	1	2	Saved	11100	4
0	0	1	0	0	Saved	11101	4
0	0	0	1	1	Saved	11110	4
0	0	0	0	0	Saved	11111	4

Fig. 11a

Input Matches				Output
D0	C1	B2	A3	Mask
1	1	1	1	1111
1	1	1	0	1110
1	1	0	1	1101
1	1	0	0	1100
1	0	1	1	1011
1	0	1	0	1010
1	0	0	1	1001
1	0	0	0	1000
0	1	1	1	0111
0	1	1	0	0110
0	1	0	1	0101
0	1	0	0	0100
0	0	1	1	0011
0	0	1	0	0010
0	0	0	1	0001
0	0	0	0	0000

Figure 11b

Combined Mask	Count
0000	0
0001	1
0010	0
0011	2
0100	0
0101	1
0110	0
0111	3
1000	0
1001	1
1010	0
1011	2
1100	0
1101	1
1110	0
1111	Count+4

Figure 11c

Output Masks	Combined Mask
M1234 | 4321 &~M1	0001
432 & ~M12	0010
43 & ~M123	0100
4 & ~M1234	1000
First valid row determines Combined Mask Output	
M-Max Count Flag 1-1st Symbol Match 2-2nd Symbol Match 3-3rd Symbol Match 4-4th Symbol Match	

Figure 11d

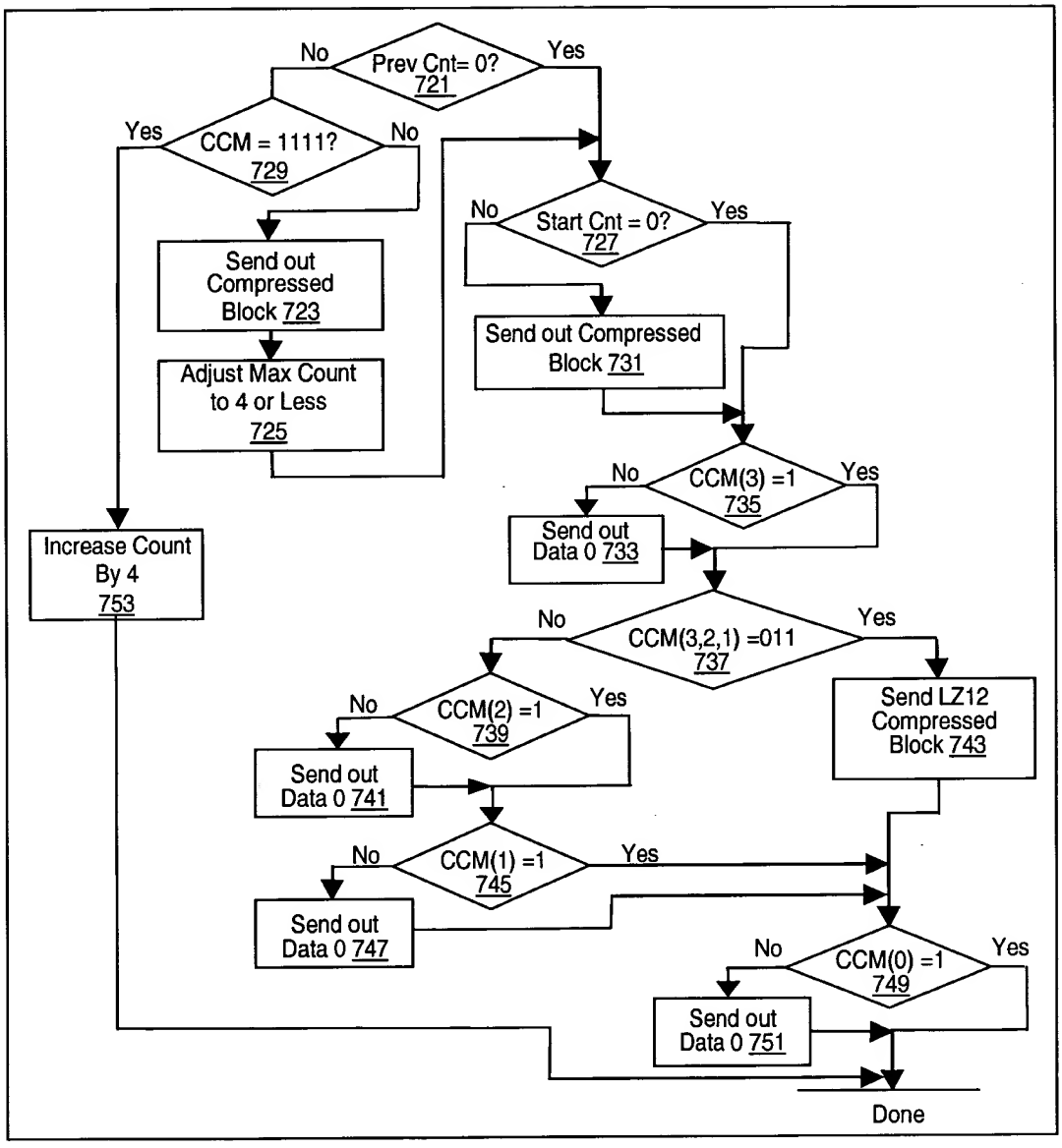

Prev Cnt= 0? 721
No
Yes
CCM = 1111? 729
Yes
No
Send out Compressed Block 723
Adjust Max Count to 4 or Less 725
Increase Count By 4 753
Start Cnt = 0? 727
No
Yes
Send out Compressed Block 731
CCM(3) =1 735
No
Yes
Send out Data 0 733
CCM(3,2,1) =011 737
No
Yes
CCM(2) =1 739
No
Yes
Send out Data 0 741
Send LZ12 Compressed Block 743
CCM(1) =1 745
No
Yes
Send out Data 0 747
CCM(0) =1 749
No
Yes
Send out Data 0 751
Done

Figure 12

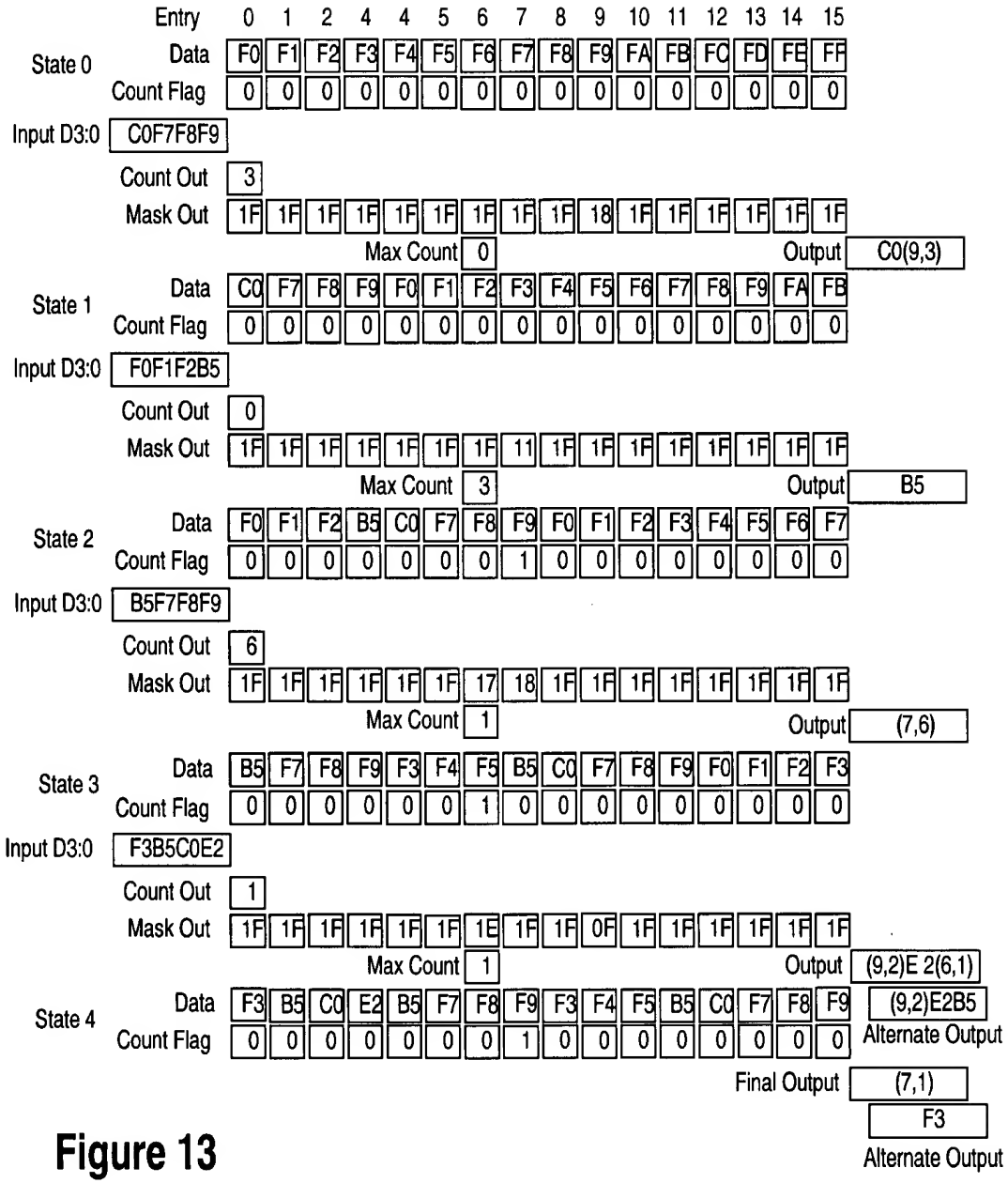


Figure 13

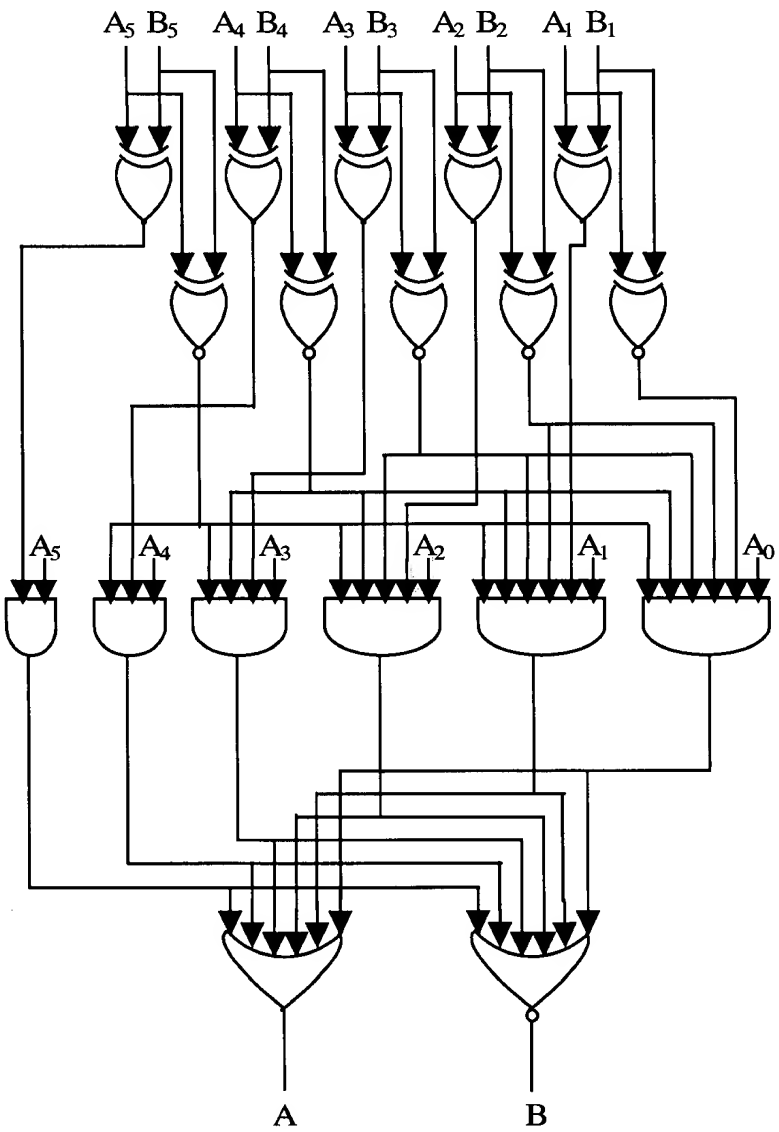


Figure 14

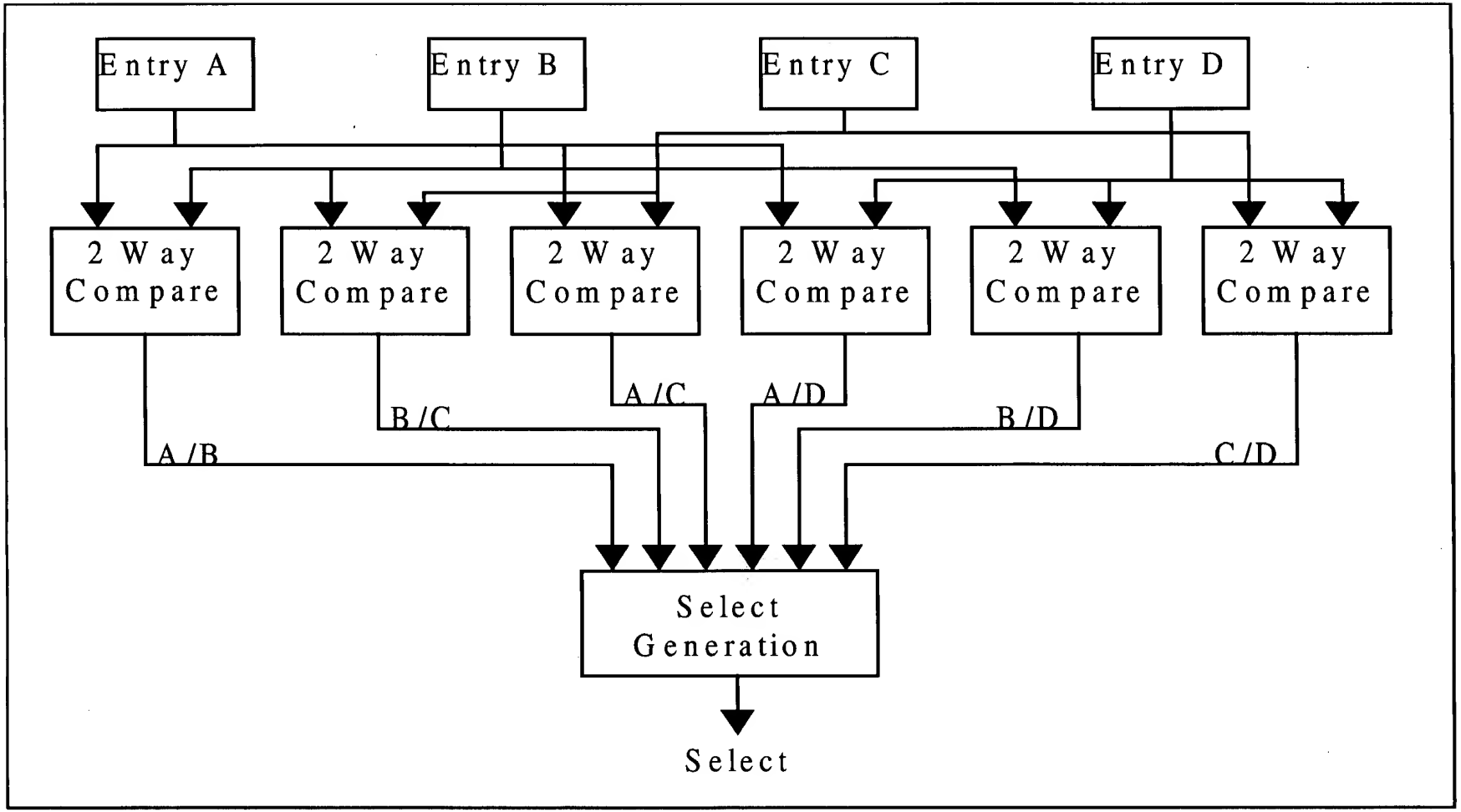


Figure 15

A/B	B/C	C/D	D/A	A/C	B/D	Output
0	X	X	1	0	X	A
1	0	X	X	X	0	B
X	1	0	X	1	X	C
X	X	1	0	X	1	D

Figure 16

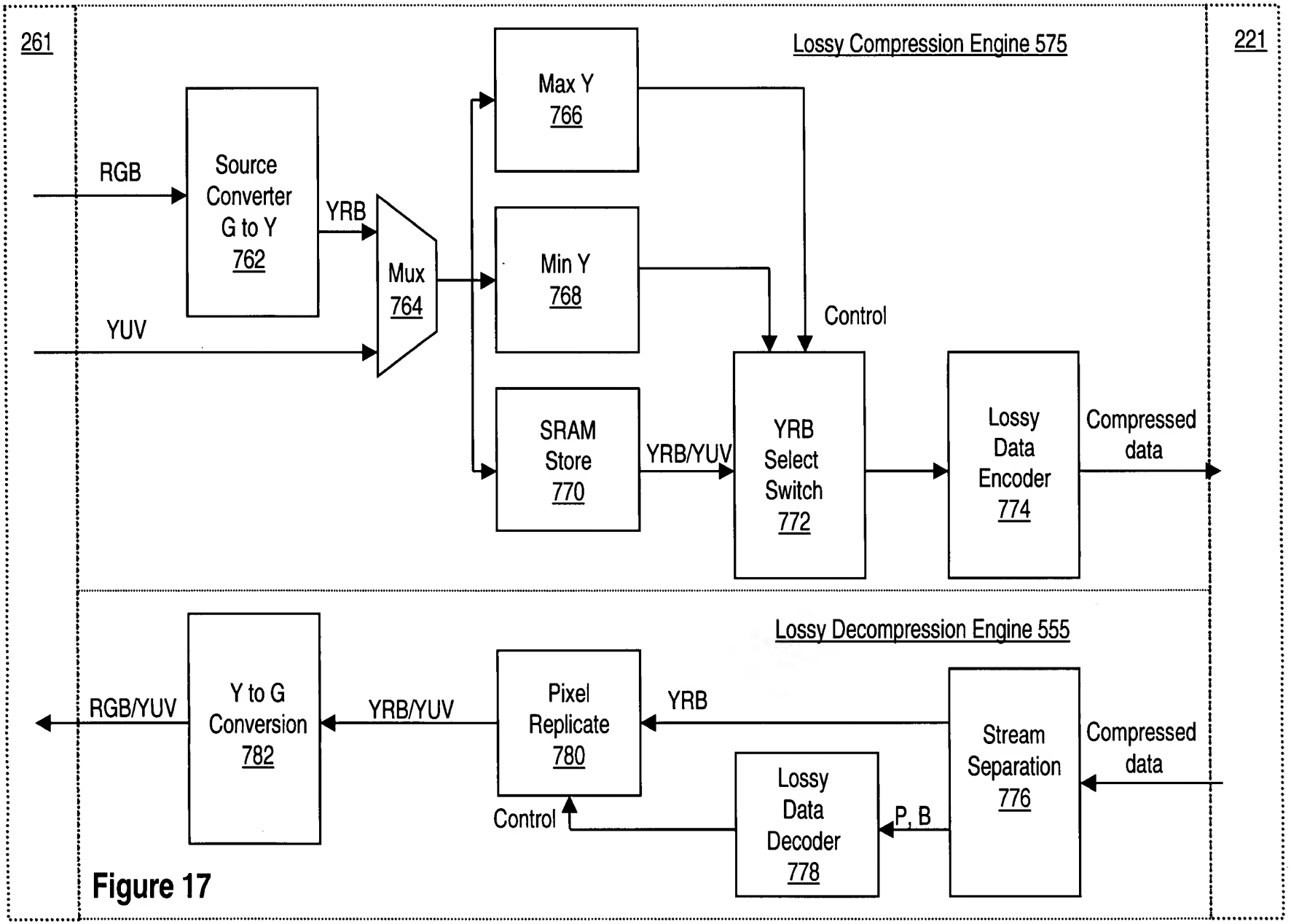


Figure 17

Ymax = Ymin	1 color	Ymax	Ymax	Rmax	Bmax	11			3 Bytes
		6 bits	6 bits	5 bits	5 bits	2 bits			
Ymax != Ymin	2 colors	Ymax	Ymin	Rmax	Rmin	Bmax	Bmin	P bits	6 Bytes
		6 bits	6 bits	5 bits	5 bits	5 bits	5 bits	16 bits	
Ymax != Ymin	>2 colors	Ymin	Ymax	Rmax	Rmin	Bmax	Bmin	P bits	8 Bytes
		6 bits	6 bits	5 bits	5 bits	5 bits	5 bits	32 bits	

Figure 18

Ymax = Ymin	Amax = Amin = 0x00	1 color	Ymax	Ymax	Rmax	Bmax	00					3 Bytes
			6 bits	6 bits	5 bits	5 bits	2 bits					
Ymax = Ymin	Amax = Amin = 0xFF	1 color	Ymax	Ymax	Rmax	Bmax	11					3 Bytes
			6 bits	6 bits	5 bits	5 bits	2 bits					
Ymax = Ymin	Amax = Amin != 00 or FF	1 color	Ymax	Ymax	Rmax	Bmax	01	Amax	Amin			4/5 Bytes
			6 bits	6 bits	5 bits	5 bits	2 bits	4/8 bits	4/8 bits			
Ymax = Ymin	Amax != Amin	1 color	Ymax	Ymax	Rmax	Bmax	01	Amax	Amin	P bits		6/7 Bytes
		2 Alphas	6 bits	6 bits	5 bits	5 bits	2 bits	4/8 bits	4/8 bits	16 bits		
Ymax = Ymin	Amax != Amin	1 color	Ymax	Ymax	Rmax	Bmax	10	Amax	Amin	P bits		8/9 Bytes
		>2 Alphas	6 bits	6 bits	5 bits	5 bits	2 bits	4/8 bits	4/8 bits	32 bits		
Ymax != Ymin	X	2 colors	Ymax	Ymin	Rmax	Rmin	Bmax	Bmin	Amax	Amin	P bits	7/8 Bytes
			6 bits	6 bits	5 bits	5 bits	5 bits	5 bits	4/8 bits	4/8 bits	16 bits	
Ymax != Ymin	X	>2 colors	Ymin	Ymax	Rmax	Rmin	Bmax	Bmin	Amax	Amin	P bits	9/10 Bytes
			6 bits	6 bits	5 bits	5 bits	5 bits	5 bits	4/8 bits	4/8 bits	32 bits	

Figure 19

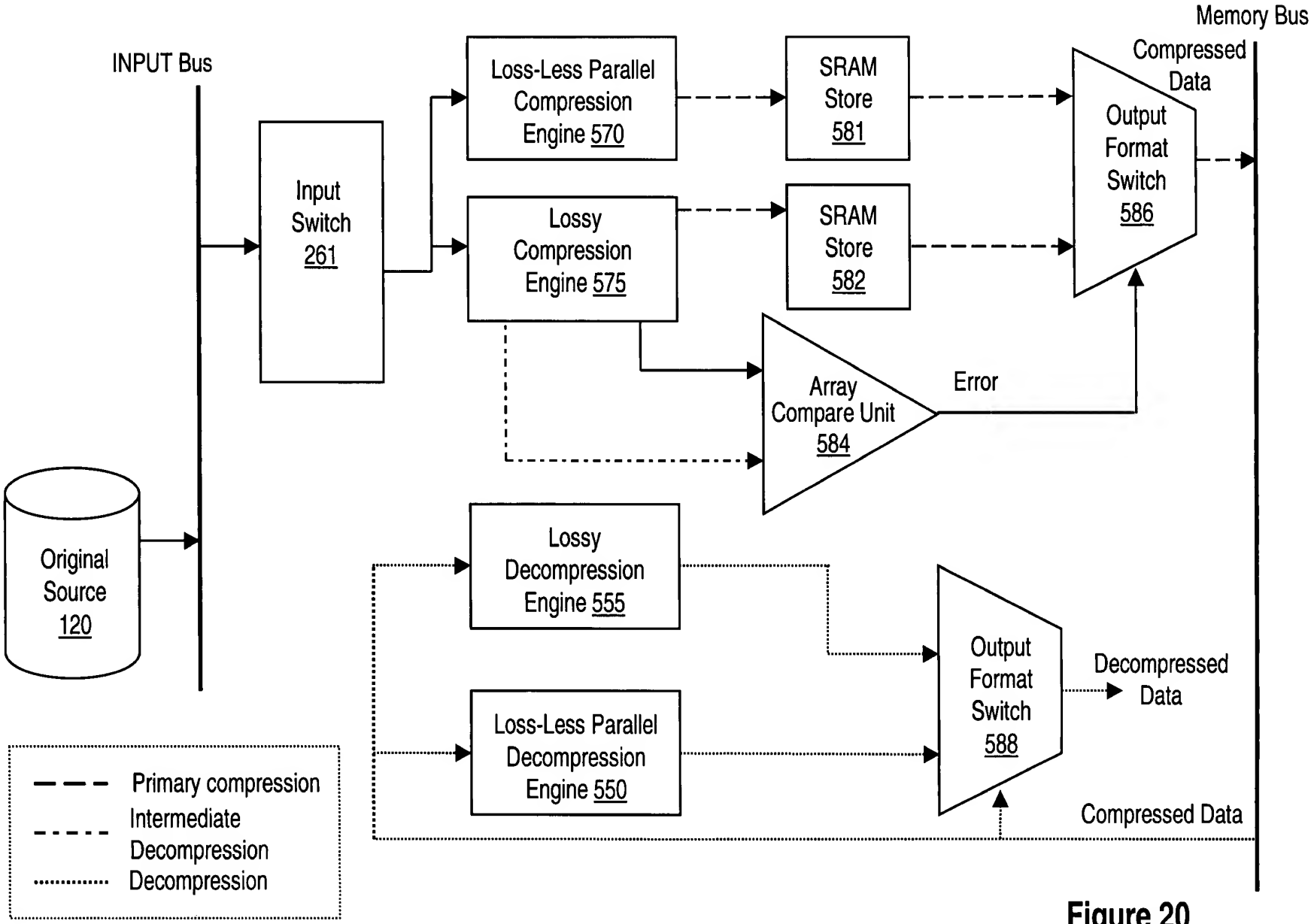


Figure 20

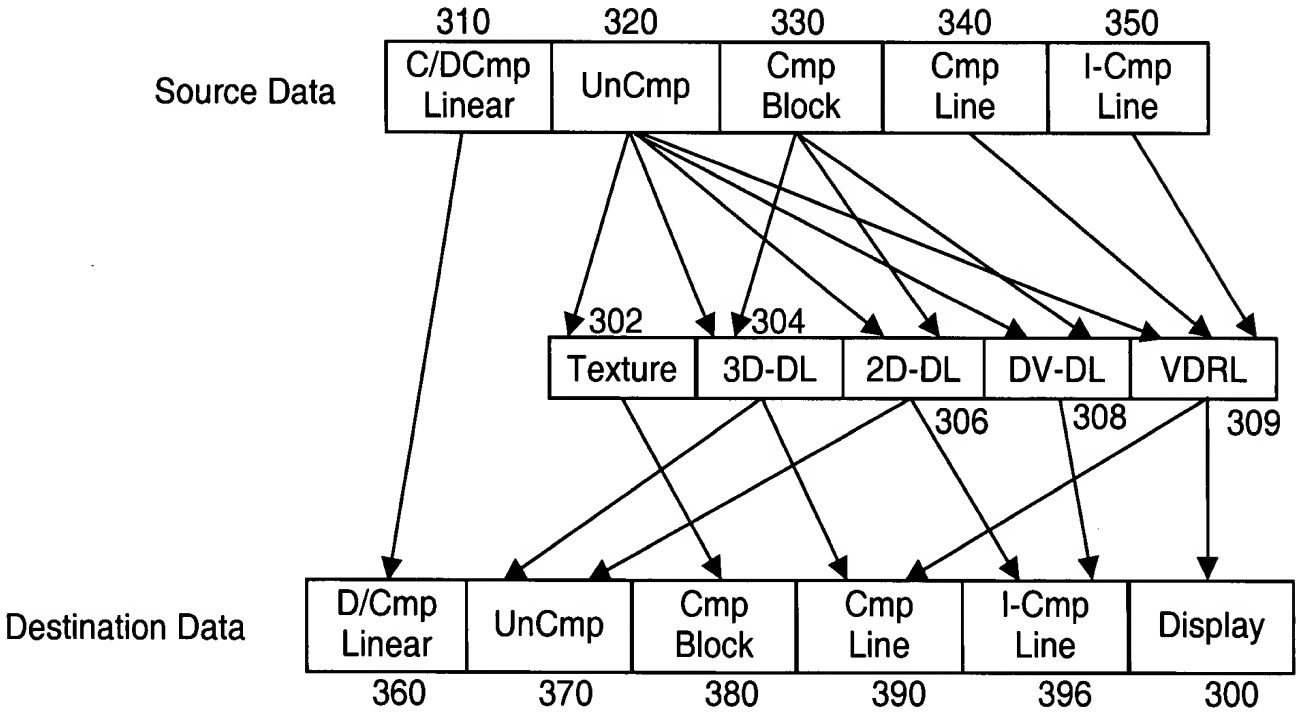


Figure 21

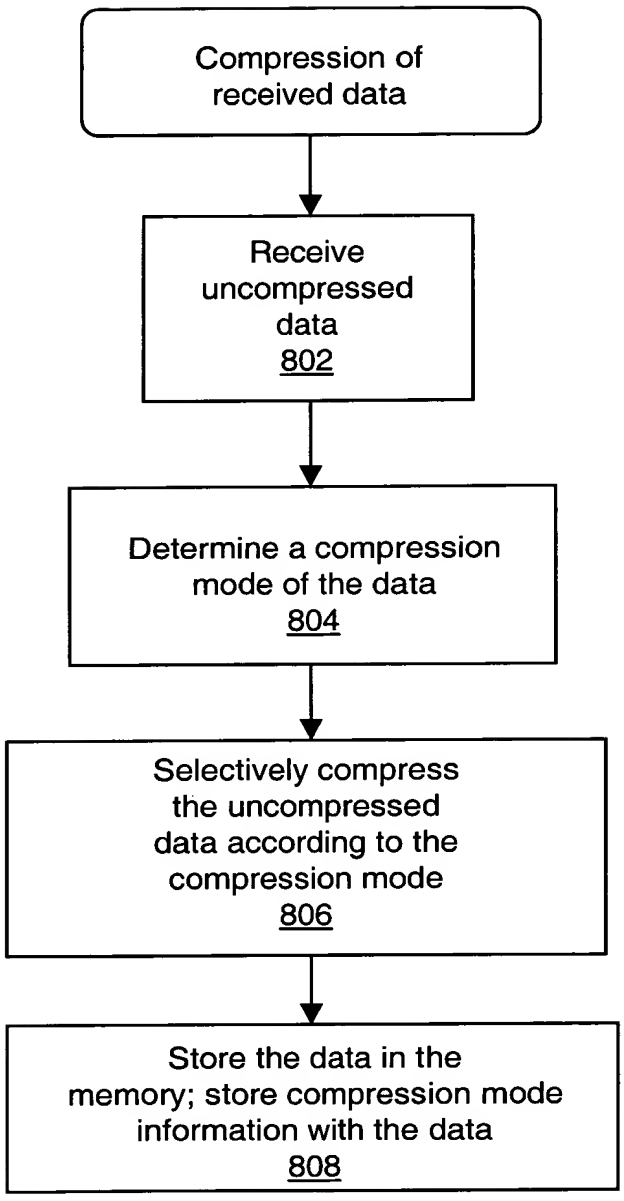


Figure 22

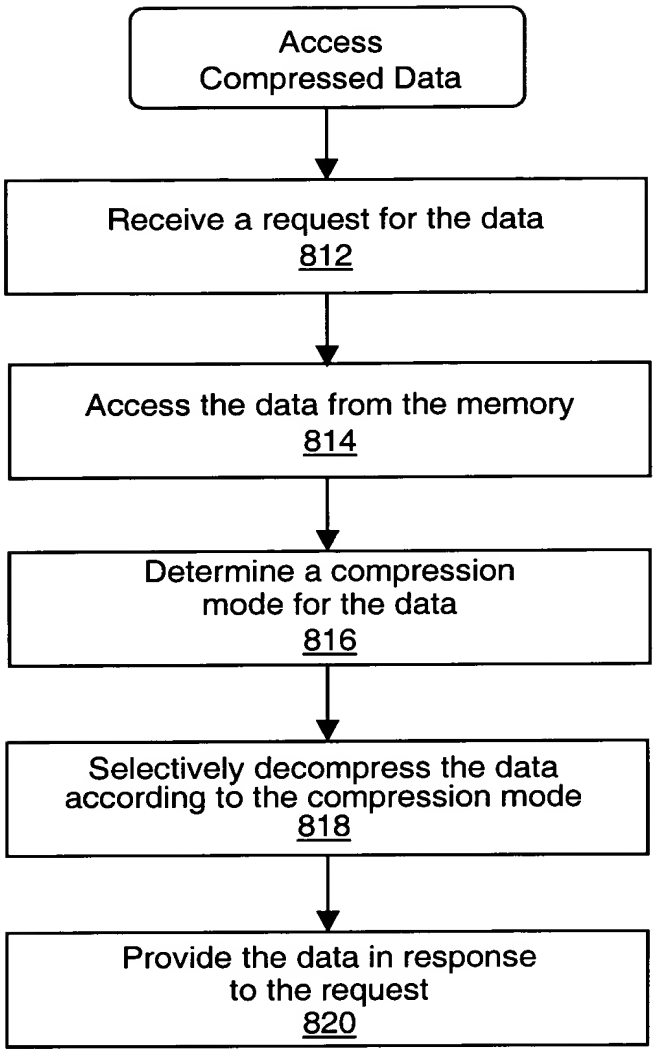

Access
Compressed Data
Receive a request for the data
812
Access the data from the memory
814
Determine a compression
mode for the data
816
Selectively decompress the data
according to the compression mode
818
Provide the data in response
to the request
820

Figure 23

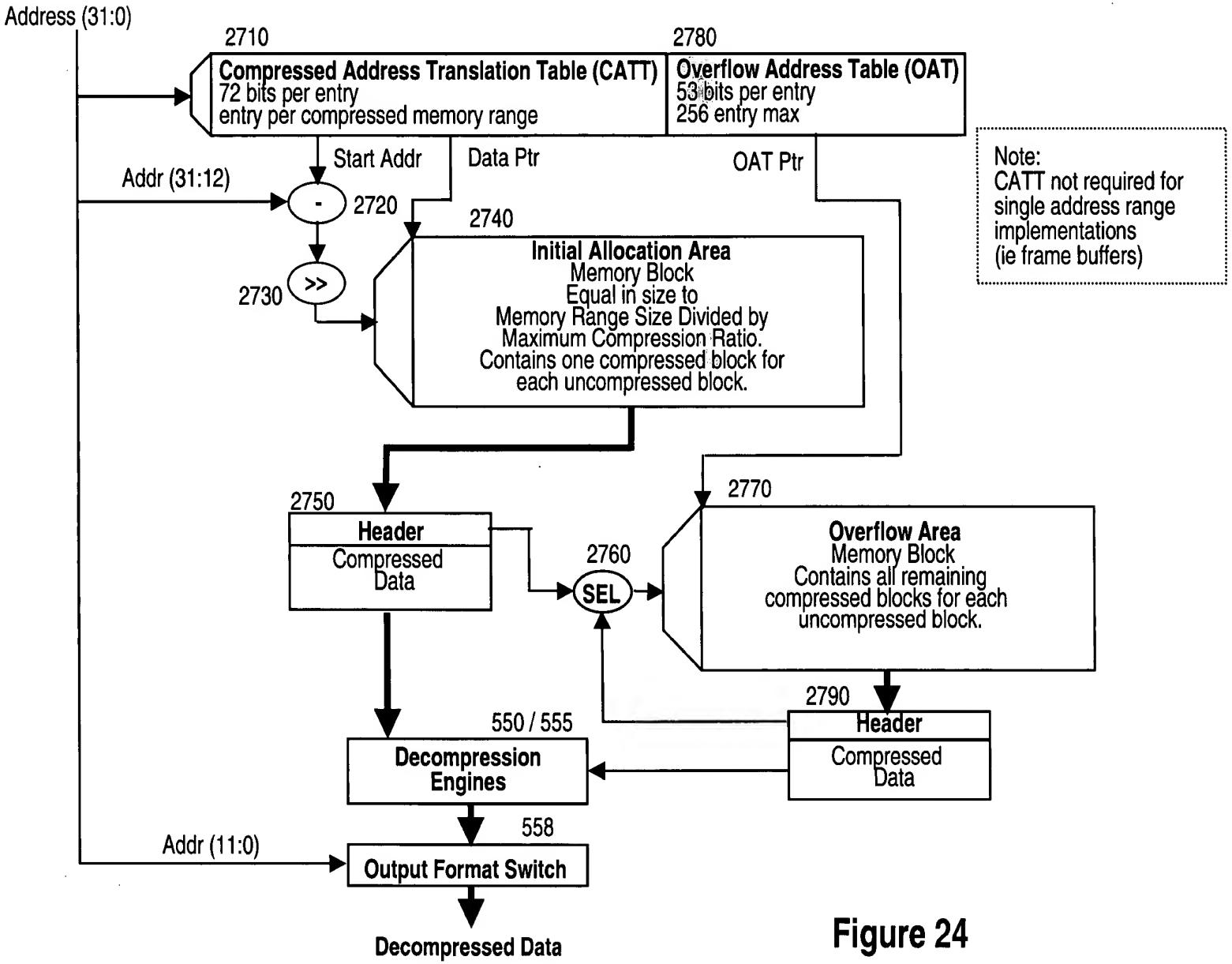


Figure 24

Memory Allocation Fields

Compressed Address Translation Table (CATT) – 128 Entry Design Limit				
Starting Addr	Ending Addr	Type	Data Ptr	OAT Ptr
20 bits	20 bits	4 bits	20 bits	8 bits
4GB Addressability		Compressed		
4K Boundry	4K Boundry	Blk Size	4K Boundry	4K Boundry

Overflow Address Table (OAT) – 256 Entry Max			
Overflow Ptr	Next Block Ptr	Next OAT Ptr	Next OAT Valid
20 bits	24 bits	8 bits	1 bit
4 GB Addressability		Points to next entry	
4K Boundry		in this table	

Initial Header Description		
Value	# of bits	Meaning
0	1	Last Block/Unused
10 A (20 bits)	22	The next block is at offset A in the Overflow Area
11 IA(8+20 bits)	30	The next block is at offset A in the Overflow Area of OAT entry I

Overflow Header Description		
Value	# of bits	Meaning
00	2	Last Block/Unused
01	2	The next block follows physically after this one
10A (8 bits)	10	The next block is A blocks before this one (or after?)
110A (20 bits)	23	The next block is at offset A in the Overflow Area
111 IA (8+20 bits)	31	The next block is at offset A in the Overflow Area of OAT entry I

Figure 25

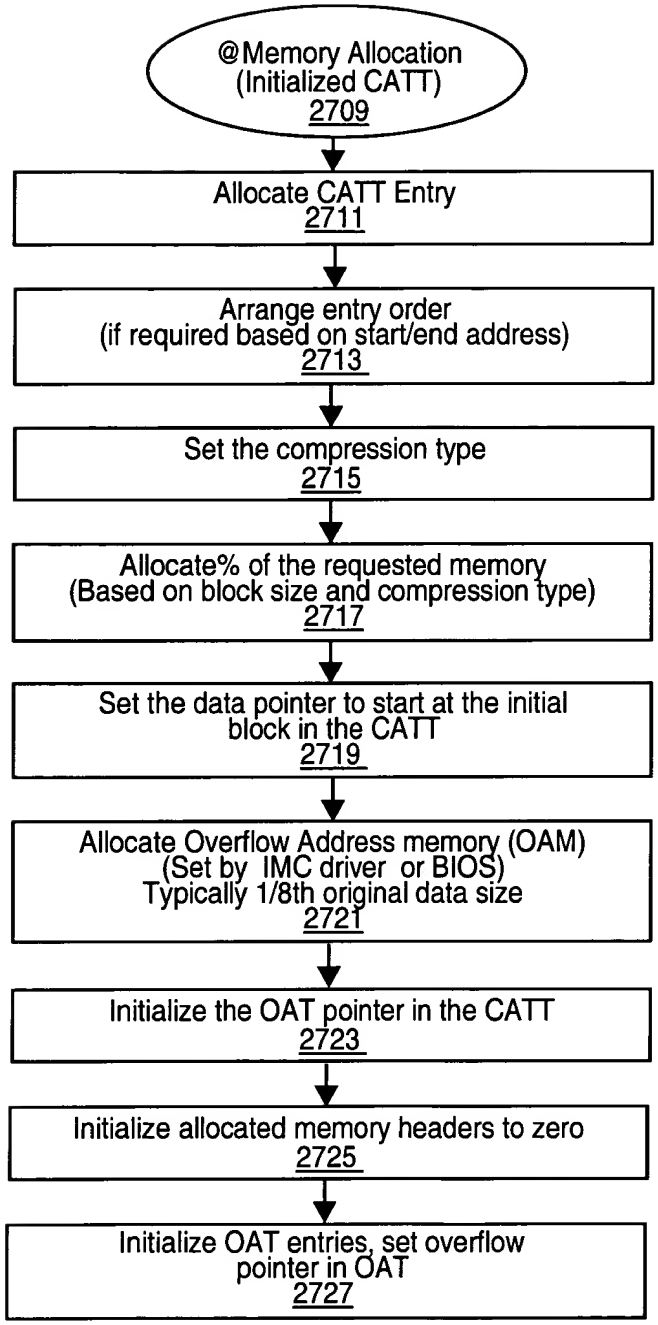


Figure 26

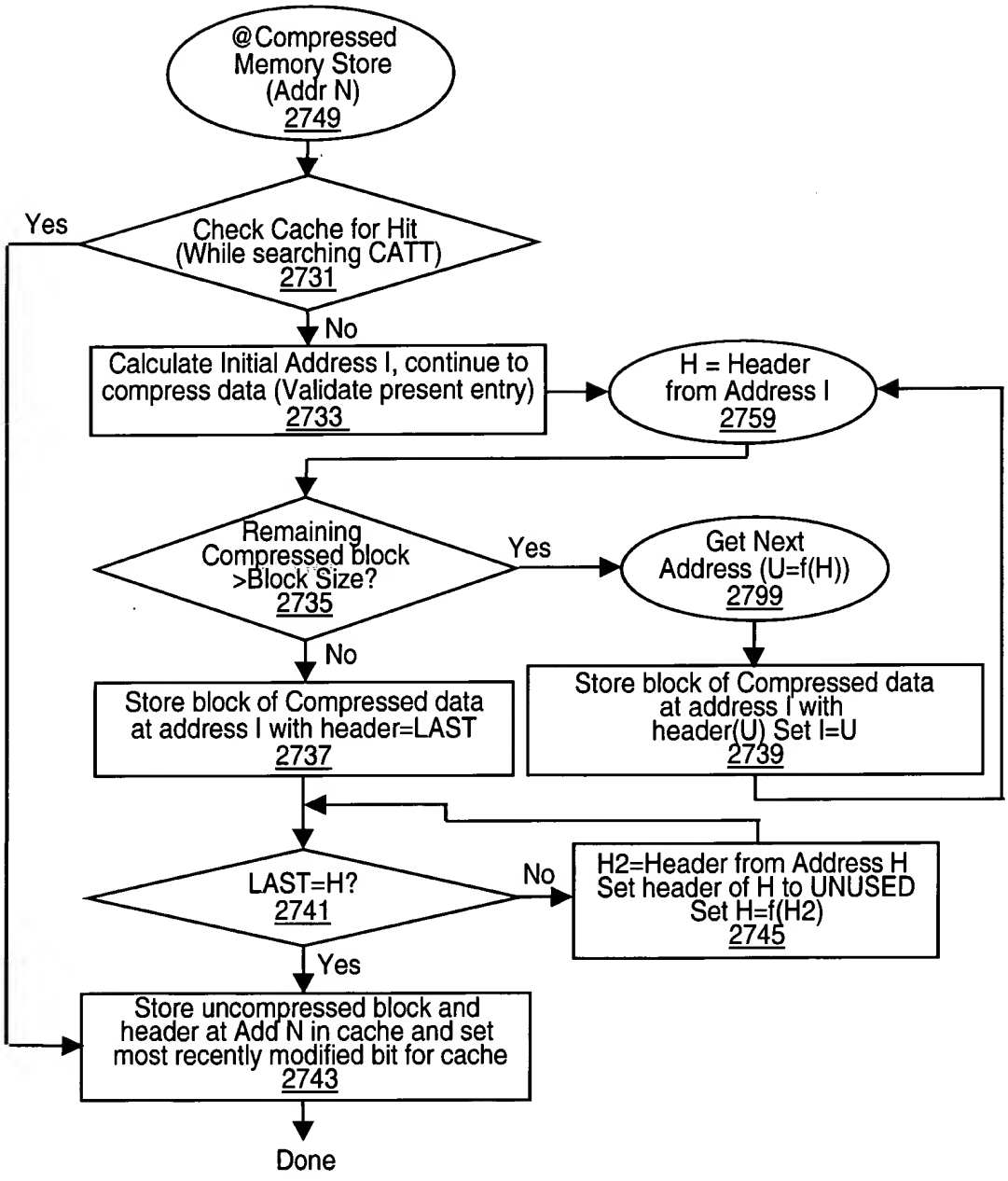


Figure 27

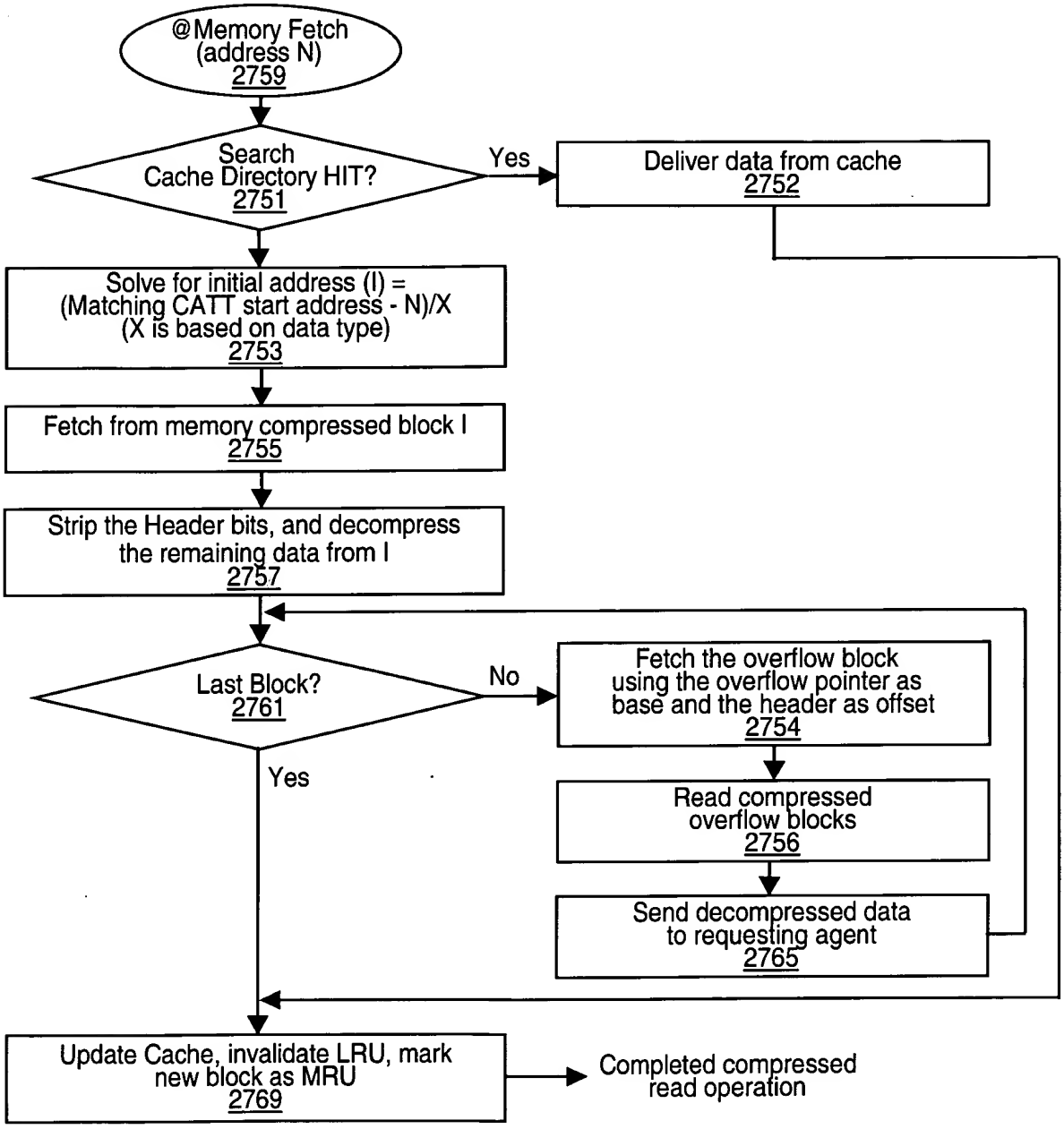


Figure 28

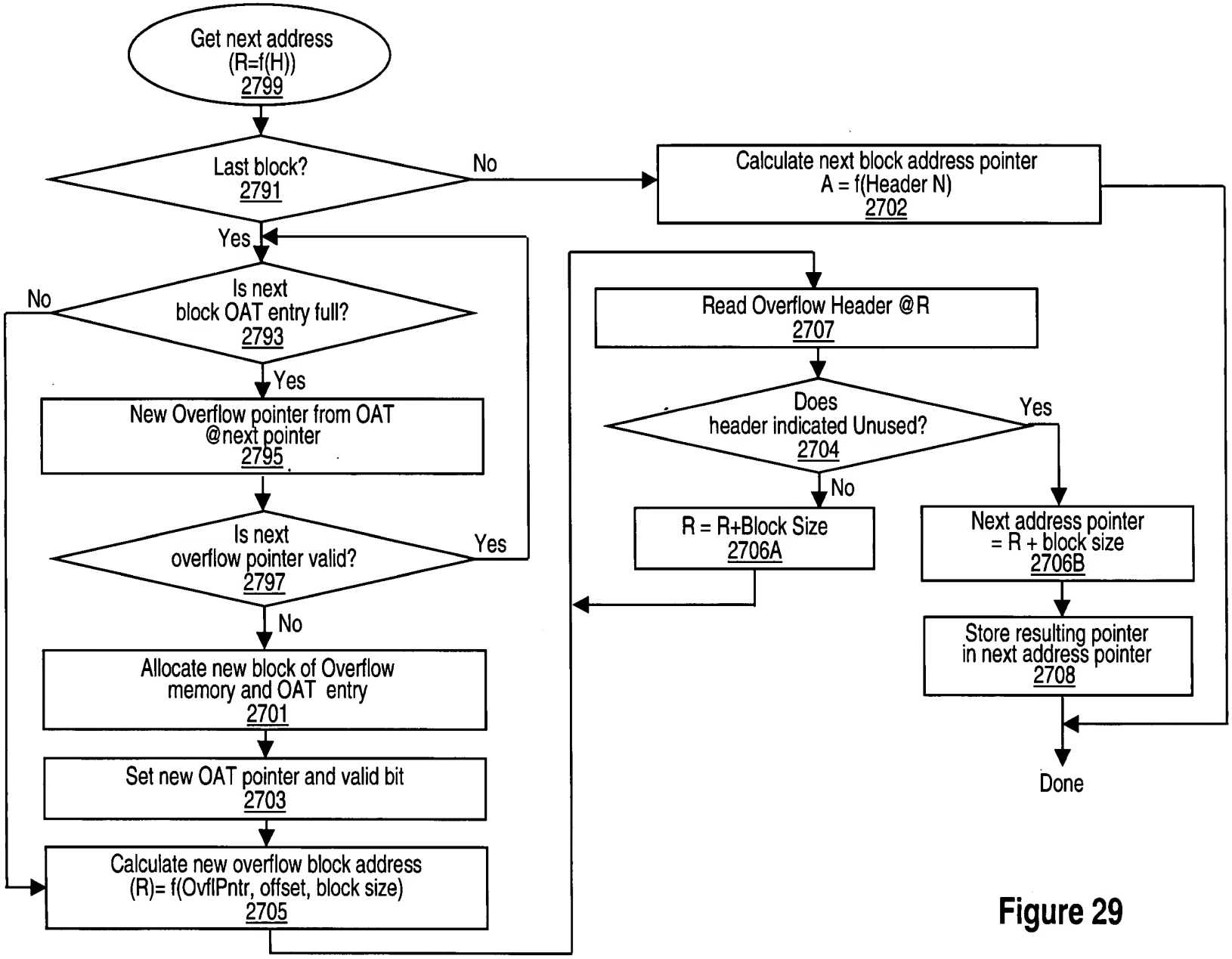
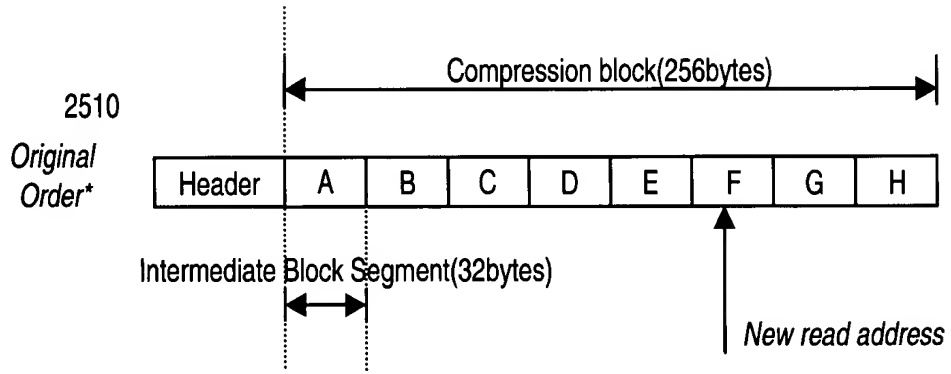


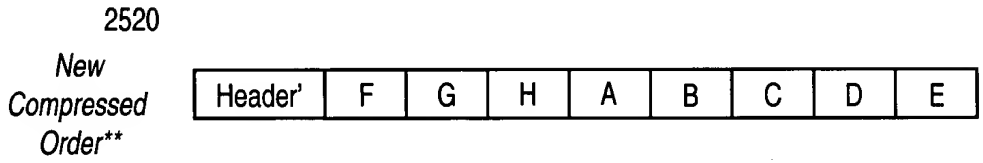
Figure 29

Uncomp Block Bytes	Type	Initial Block Size Bytes	Overflow Block Size Bytes	Max Comp Ratio (X:1)	Initial Allocation	Header w/o OF	Header w/ OF Non-Frag	Header w/ OF Fragmented
4096	8	256	64	16	6%	0.0%	0.4%	4.1%
2048	7	128	64	16	6%	0.1%	0.5%	4.2%
1024	6	64	64	16	6%	0.2%	0.6%	4.3%
512	5	64	64	8	13%	0.2%	0.9%	4.3%
256	4	64	64	4	25%	0.2%	1.4%	4.3%
128	3	32	32	4	25%	0.4%	2.8%	8.8%
64	2	32	16	2	50%	0.4%	5.1%	13.6%
32	1	32	8	1	100%	0.4%	8.9%	11.5%

Figure 30



* Header tag field is '0' indicating in original order



** Header tag field is '5' indicating out of order

2530

1) Read compressed block from memory
2) Decompress to L3 data cache
3) Store Position tag (@F) in table
4) Re-compress block out of order
5) Attach header with tag (@F)
6) Write compressed block back to memory

Figure 31

Bytes Compressed	Flag	Index	Count	Data	**Bits Used**
0	0	-	-	8b	9
1	10	6b	-	-	8
2	1100	6b	-	-	10
3	1101	6b	-	-	10
4	1110	6b	-	-	10
5	1111000	6b	-	-	13
6	1111001	6b	-	-	13
7	1111010	6b	-	-	13
8	1111011	6b	-	-	13
9	1111100	6b	-	-	13
10	1111101	6b	-	-	13
11	1111110	6b	-	-	13
>11	1111111	6b	12b	-	25

Figure 32

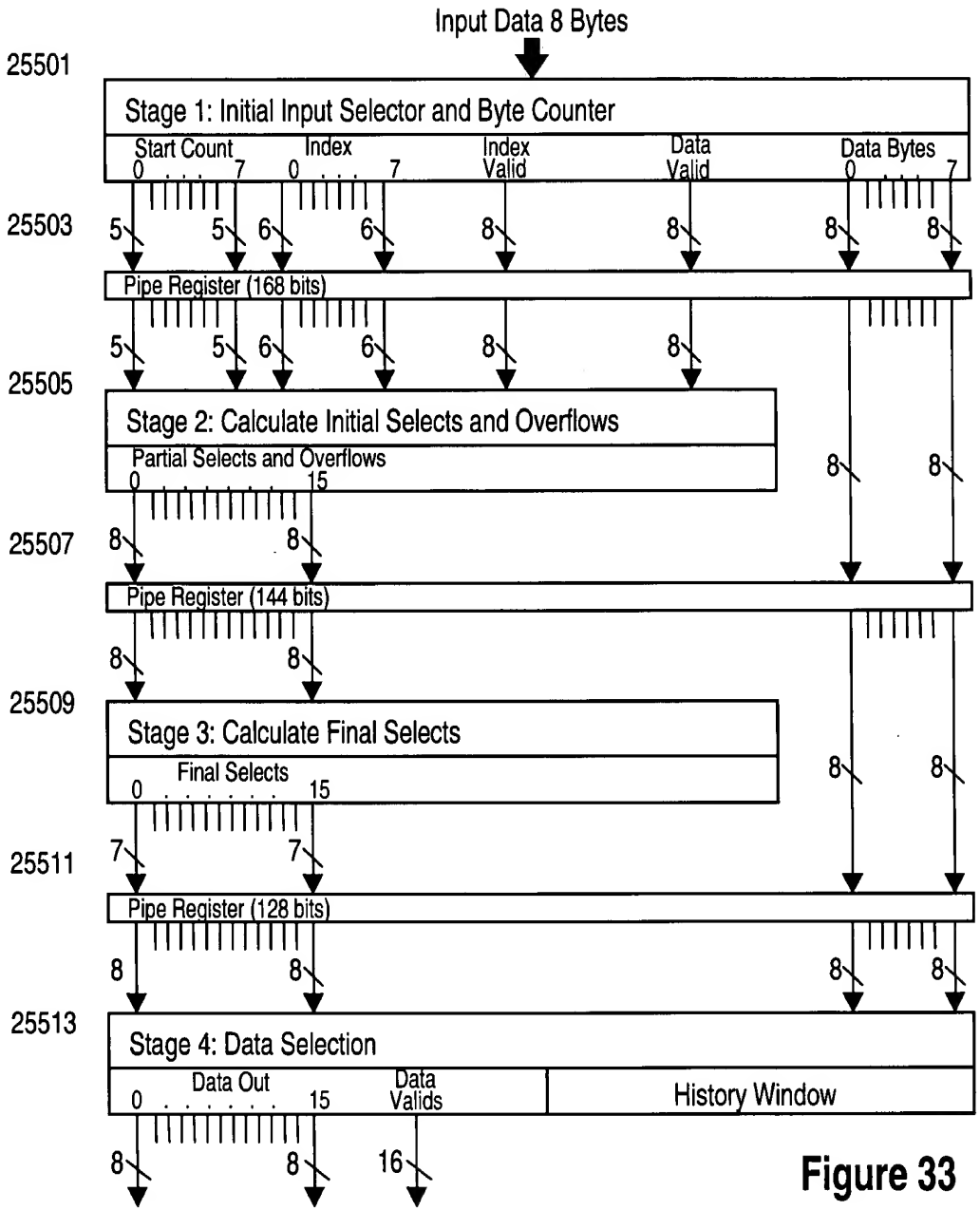


Figure 33

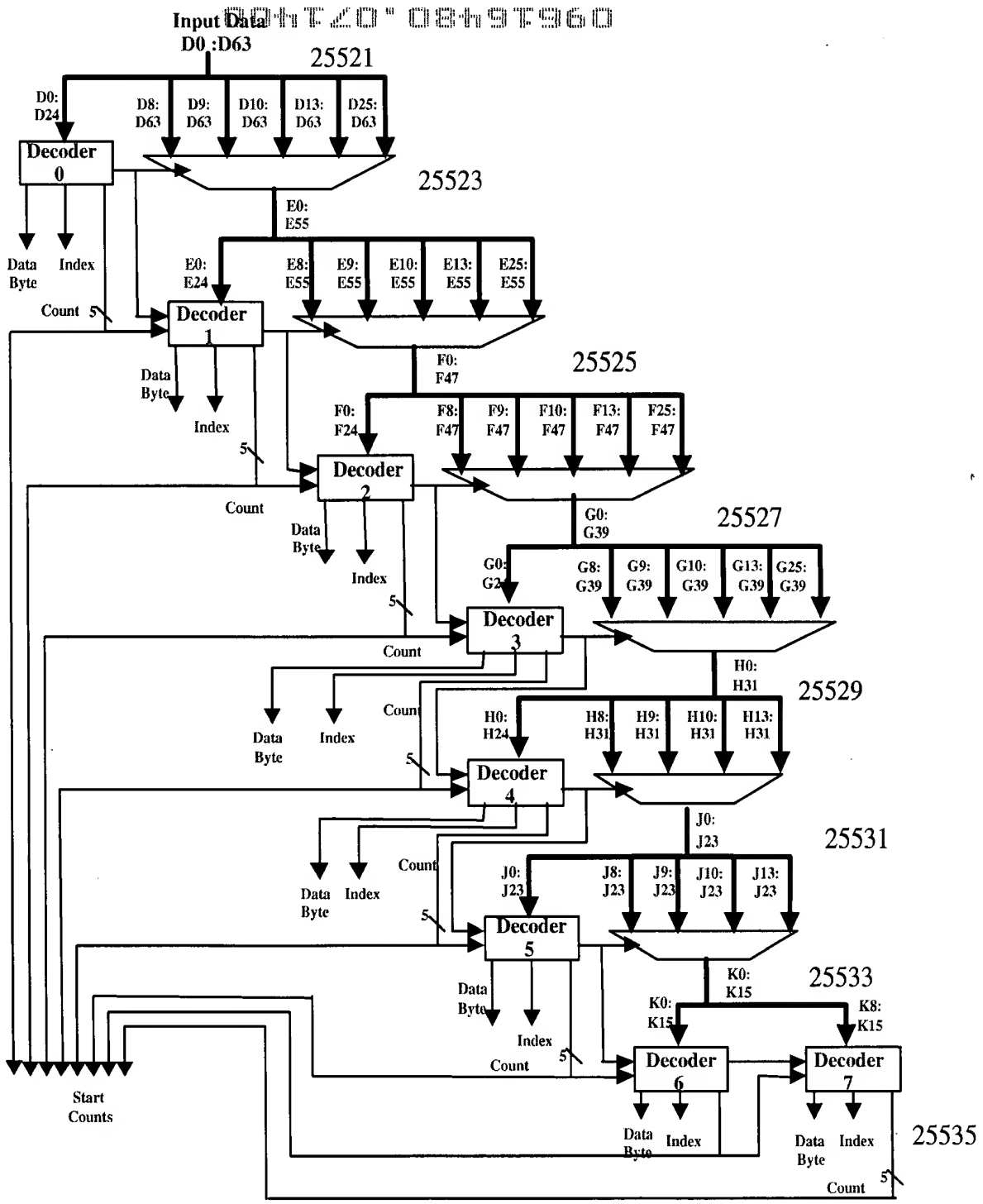

Input Data
D0 :D63
25521
D0: D24
D8: D63
D9: D63
D10: D63
D13: D63
D25: D63
Decoder 0
25523
E0: E55
Data Byte
Index
Count 5
E0: E24
E8: E55
E9: E55
E10: E55
E13: E55
E25: E55
Decoder 1
F0: F47
25525
Data Byte
Index
5
F0: F24
F8: F47
F9: F47
F10: F47
F13: F47
F25: F47
Decoder 2
Count
G0: G39
25527
Data Byte
Index
5
G0: G24
G8: G39
G9: G39
G10: G39
G13: G39
G25: G39
Decoder 3
Count
H0: H31
25529
Data Byte
Index
Count
5
H0: H24
H8: H31
H9: H31
H10: H31
H13: H31
Decoder 4
J0: J23
25531
Data Byte
Index
Count
5
J0: J23
J8: J23
J9: J23
J10: J23
J13: J23
Decoder 5
K0: K15
25533
Data Byte
Index
Count
5
K0: K15
K8: K15
Decoder 6
Decoder 7
Start Counts
Data Byte
Index
Data Byte
Index
25535
Count
5

Figure 34

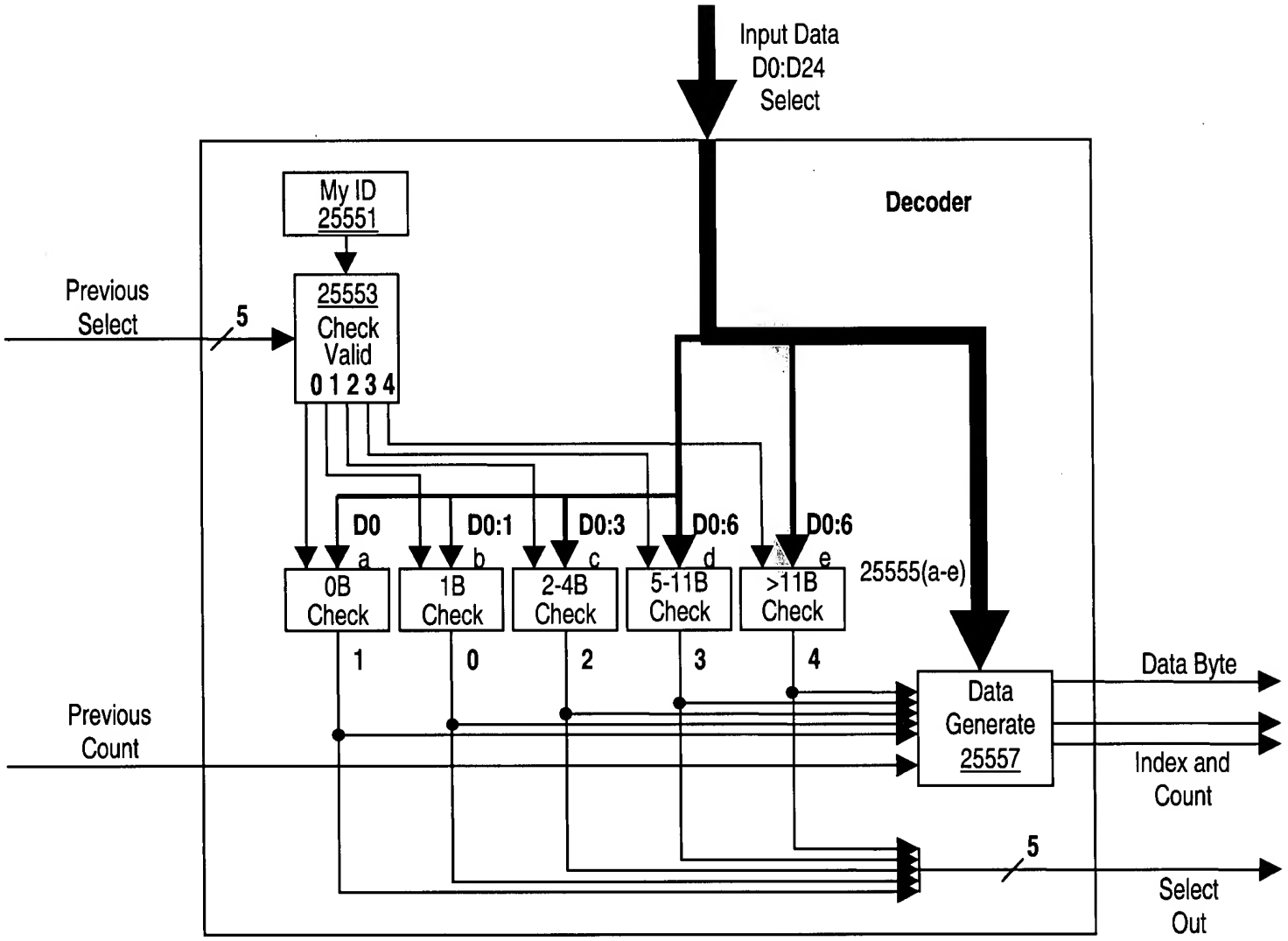

Input Data
D0:D24
Select
Decoder
My ID
25551
Previous
Select
5
25553
Check
Valid
0 1 2 3 4
D0
a
D0:1
b
D0:3
c
D0:6
d
D0:6
e
25555(a-e)
0B
Check
1B
Check
2-4B
Check
5-11B
Check
>11B
Check
1
0
2
3
4
Previous
Count
Data
Generate
25557
Data Byte
Index and
Count
5
Select
Out

Figure 35

Previous Select	10	08	04	02	01	00
My ID=01	1F	1F	1F	1F	1F	00
My ID=02	1F	1F	1F	1F	1F	00
My ID=04	1F	1F	1F	1F	1F	00
My ID=08	1F	1F	1F	1F	1E	00
My ID=10	1F	1F	1F	1F	1E	00
My ID=20	1E	1E	1E	1E	00	00
My ID=40	1E	1E	1E	1C	00	00
My ID=80	08	00	00	00	00	00

Figure 36a

Select	10	08	04	02	01	00
Data Byte	X	D1:D8	X	X	X	X
Index	D2:D7	X	D4:D9	D7:D12	D7:D12	X
Count	PC+1	PC+1	D2:D3+PC+2	D4:D6+PC+5	D13:D24+PC	X

Figure 36b

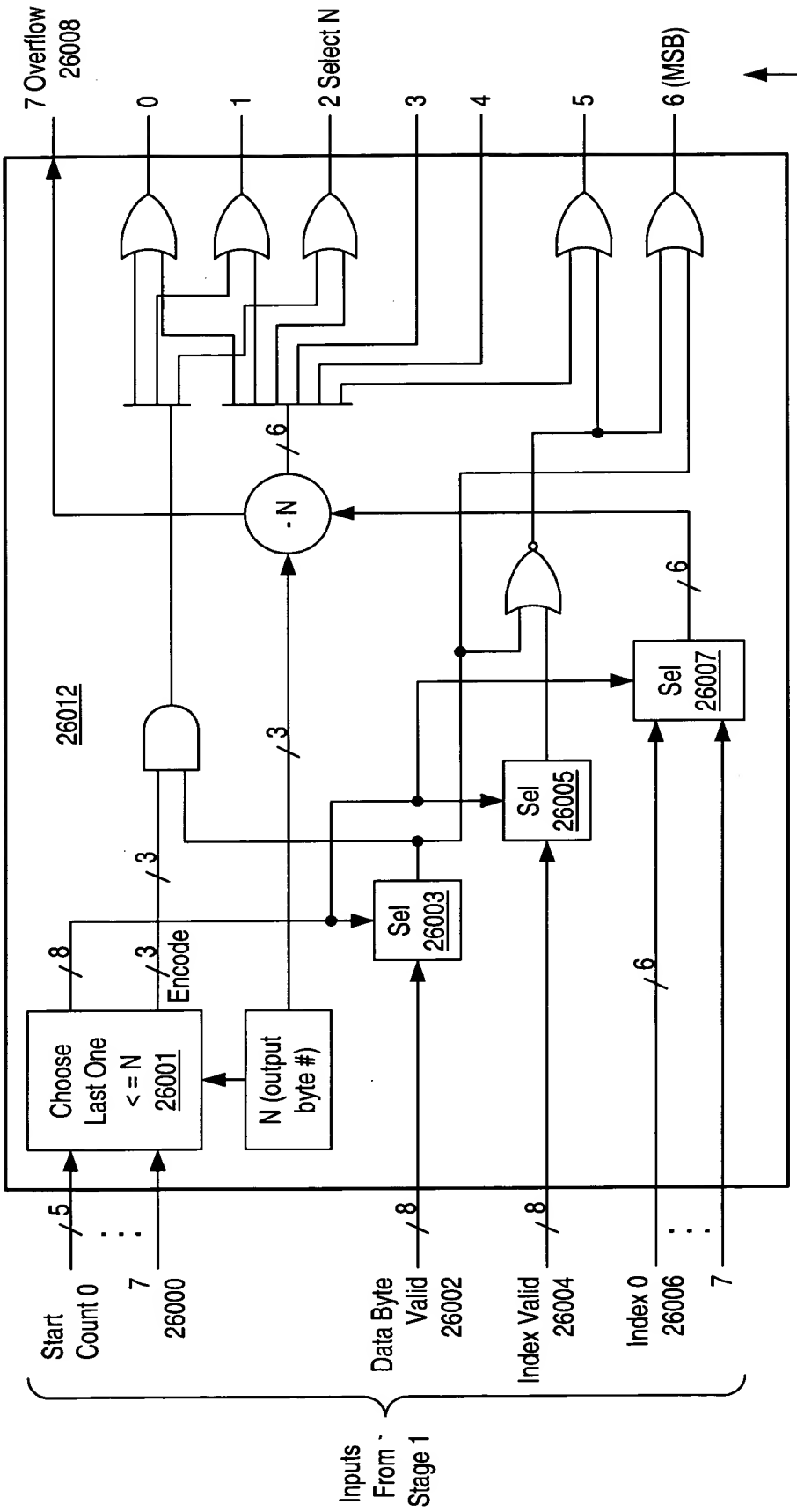

Stage 2 - Calculate Selects & Overflows
Inputs From Stage 1
Start Count 0
7
26000
Data Byte Valid
26002
Index Valid
26004
Index 0
26006
7
Choose Last One < = N
26001
N (output byte #)
Encode
Sel
26003
Sel
26005
Sel
26007
26012
- N
7 Overflow
26008
0
1
2 Select N
3
4
5
6 (MSB)
Output
26010

Figure 37

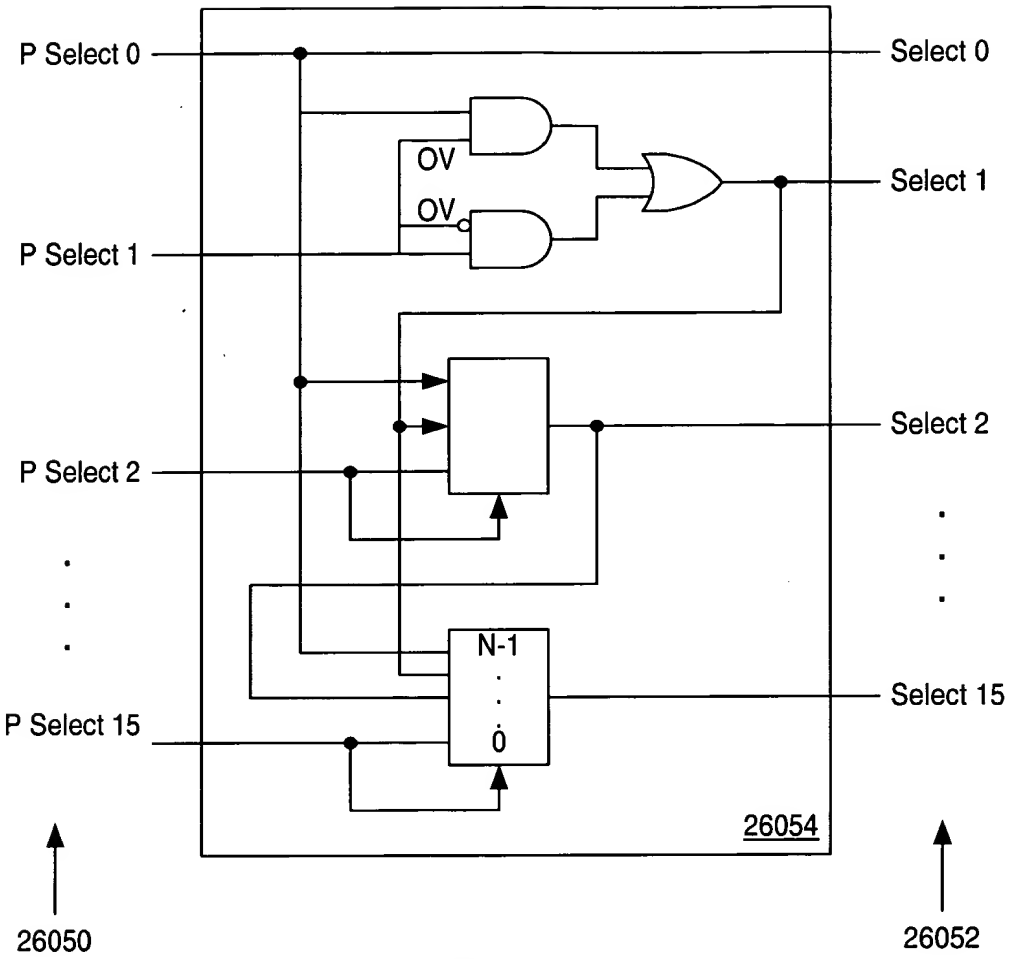


Figure 38

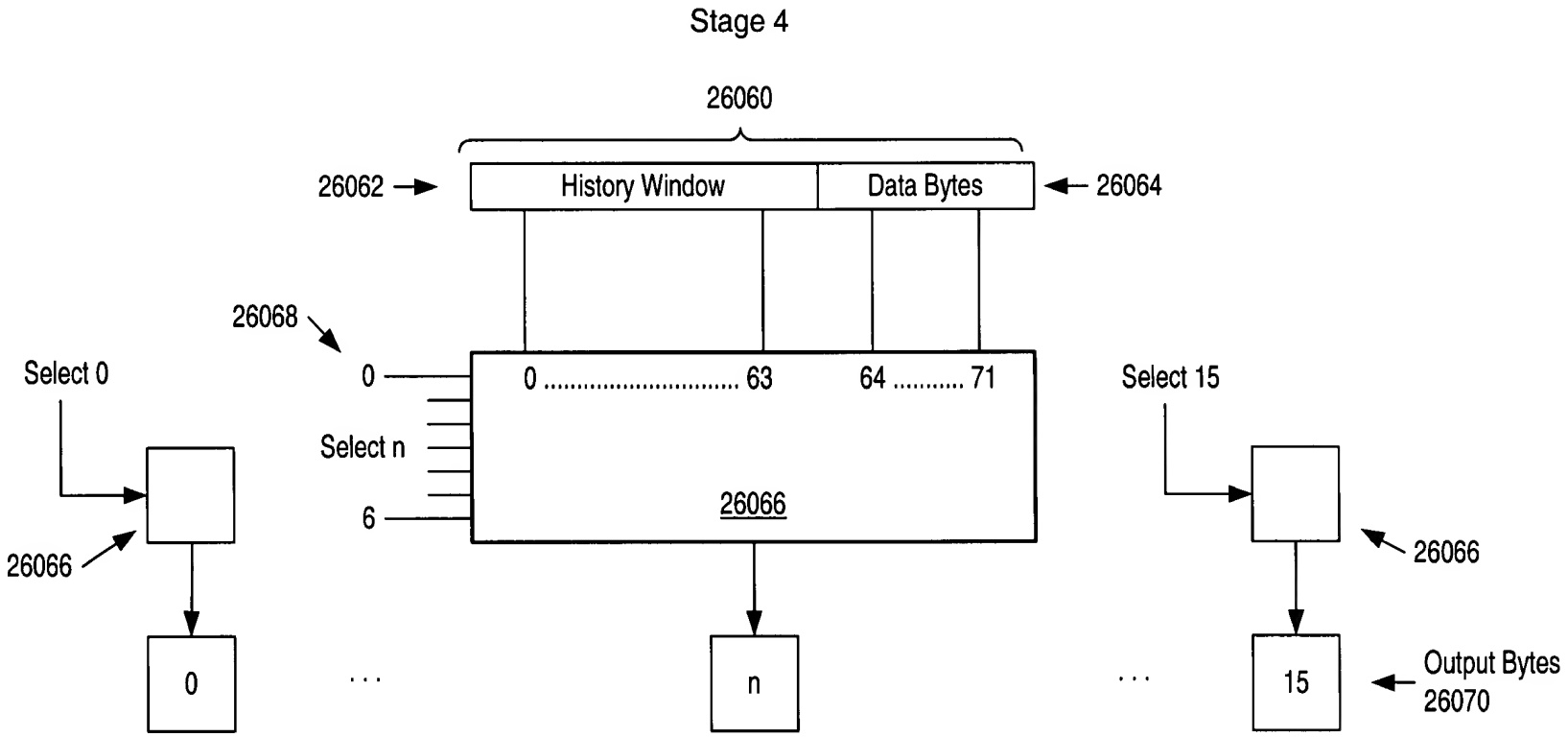
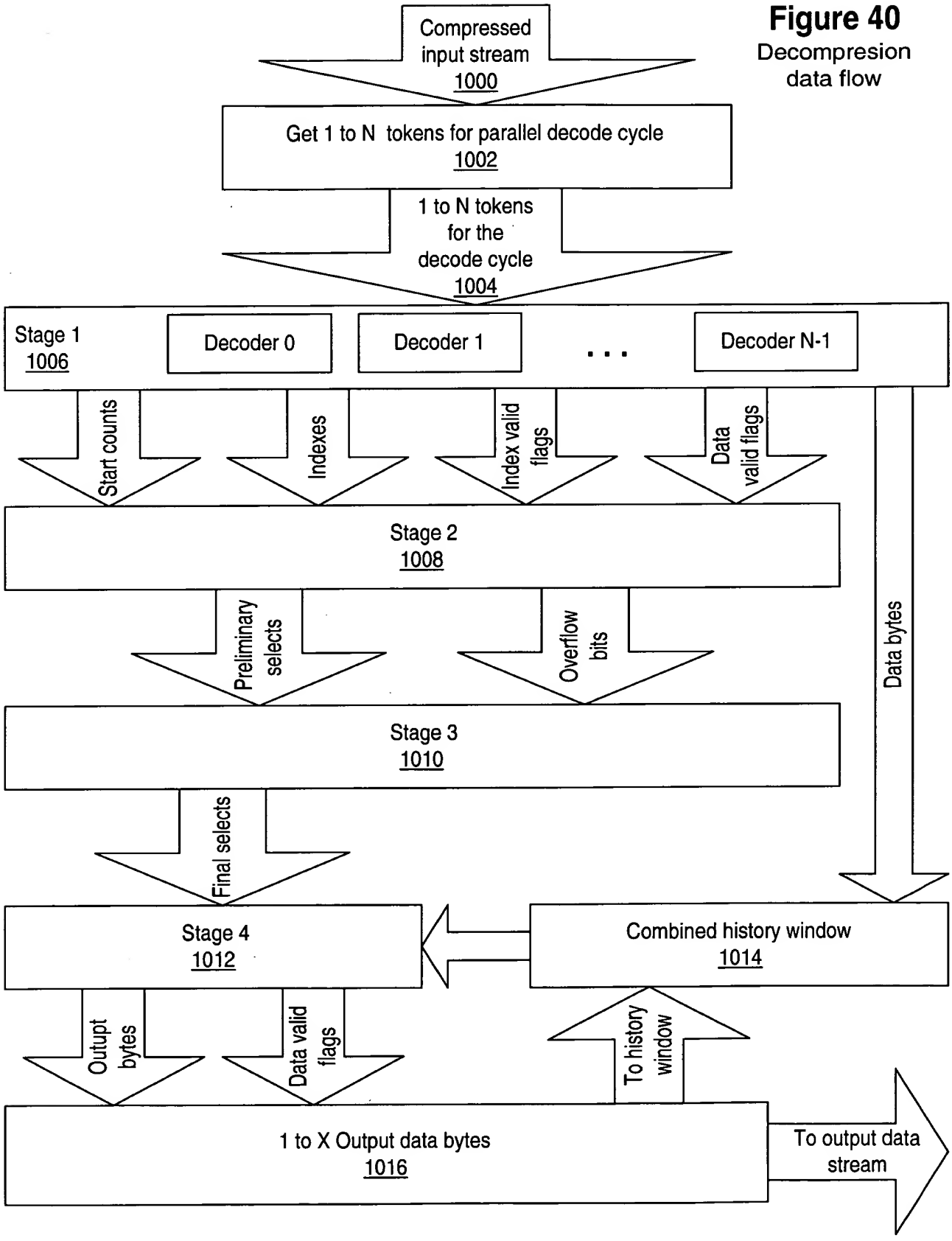
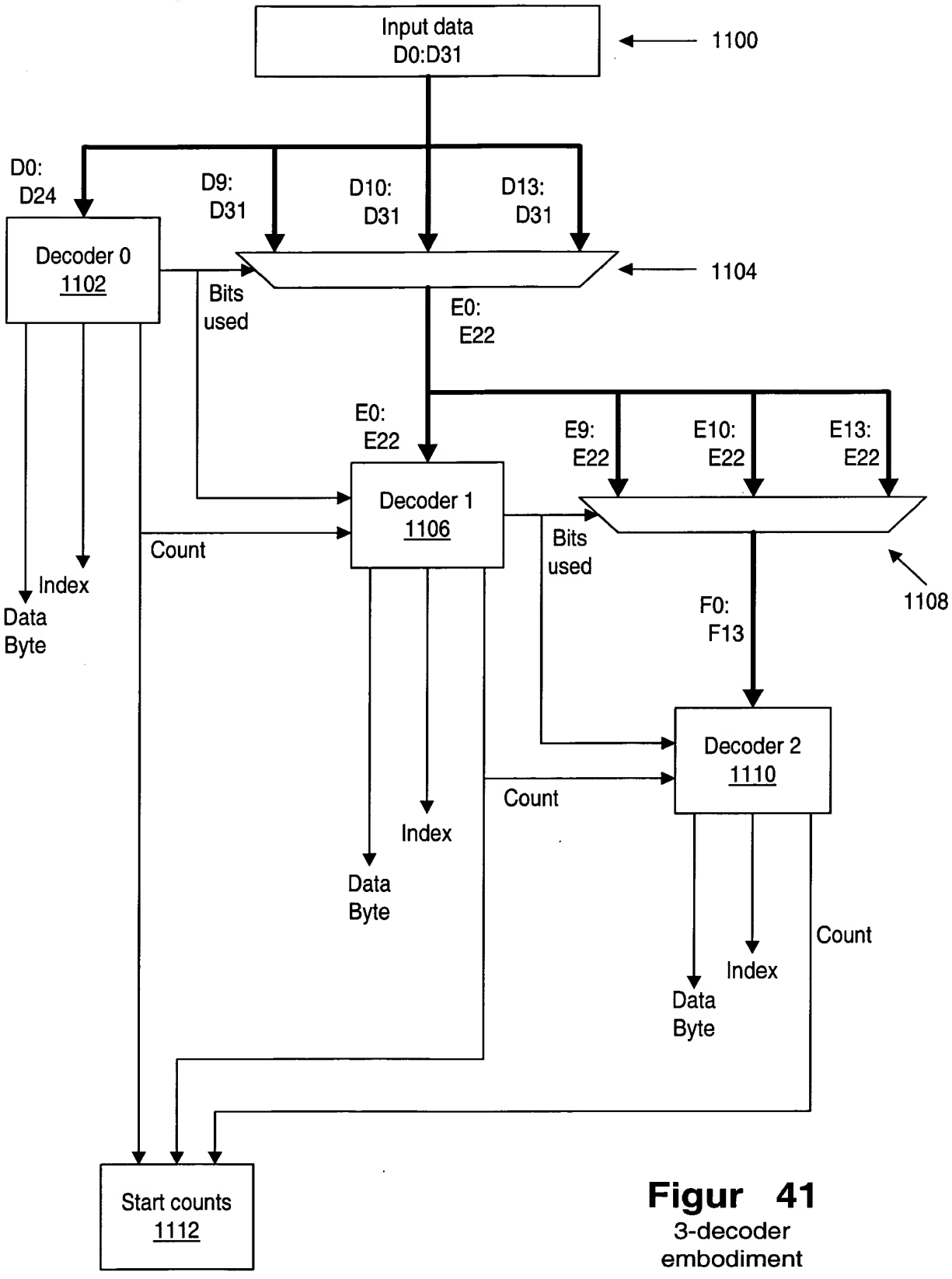


Figure 39

Figure 40
Decompresion data flow





Figur 41
3-decoder
embodiment

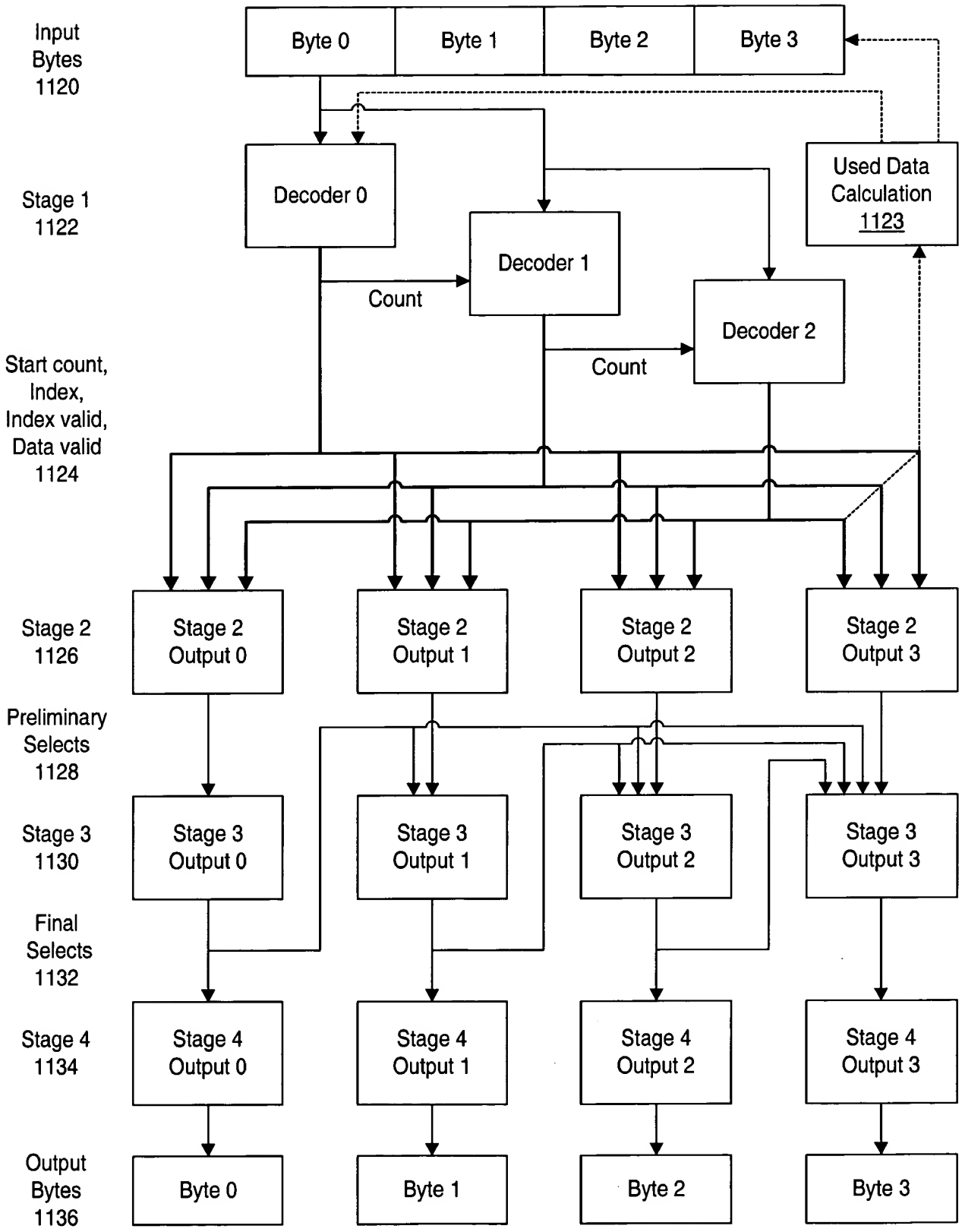


Figure 42a

- 4 input bytes, 3 decoders, 4 output bytes

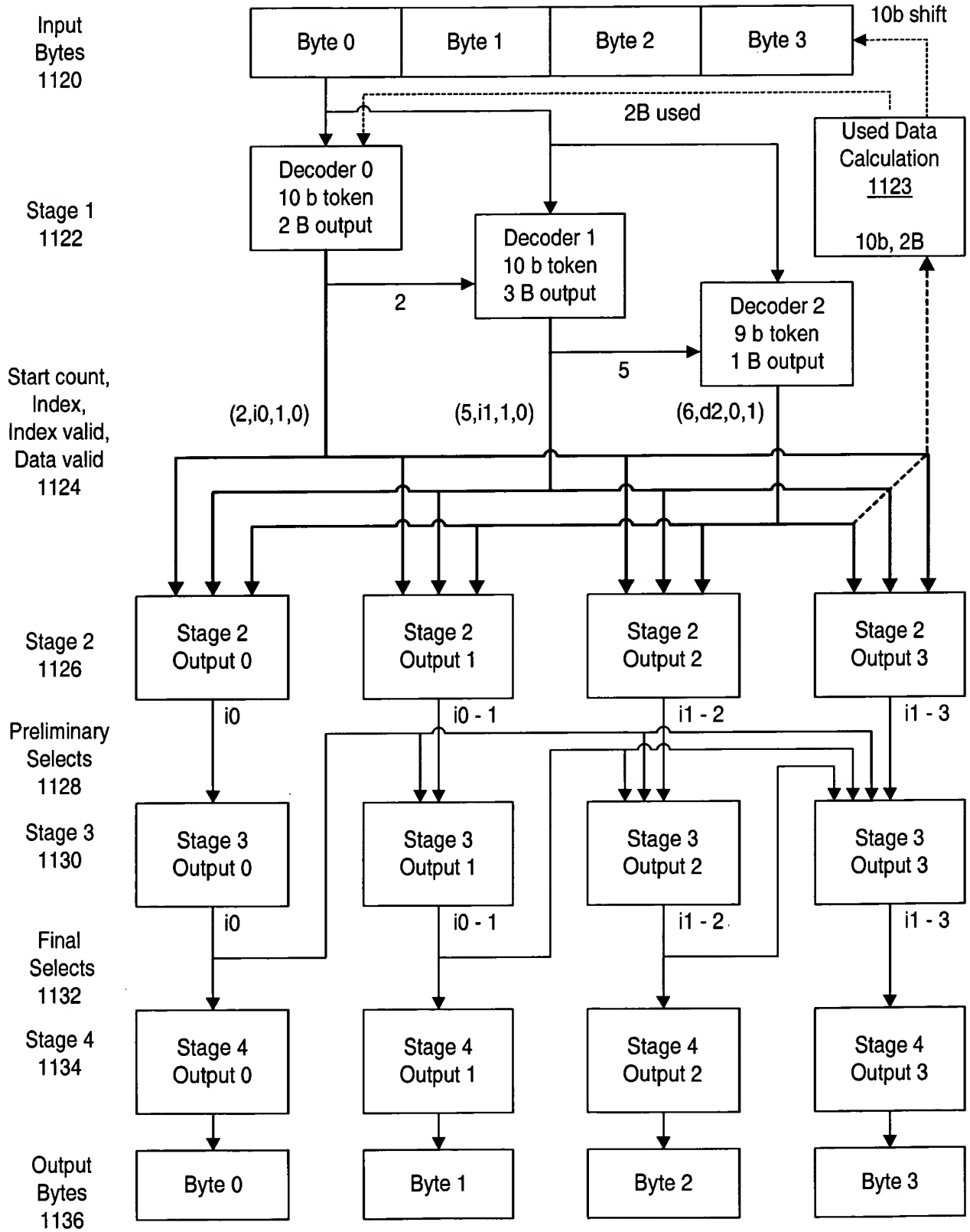


Figure 42b

- example using 4 input bytes, 3 decoders, 4 output bytes

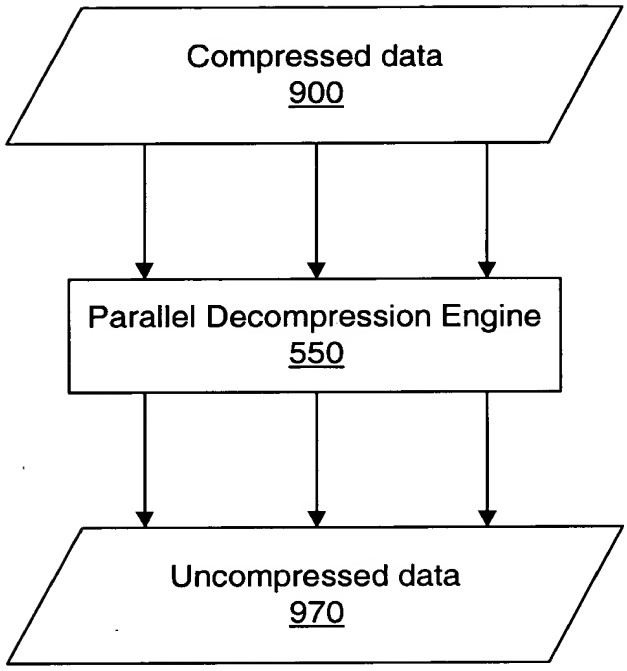


Figure 43a
Decompresion
flowchart

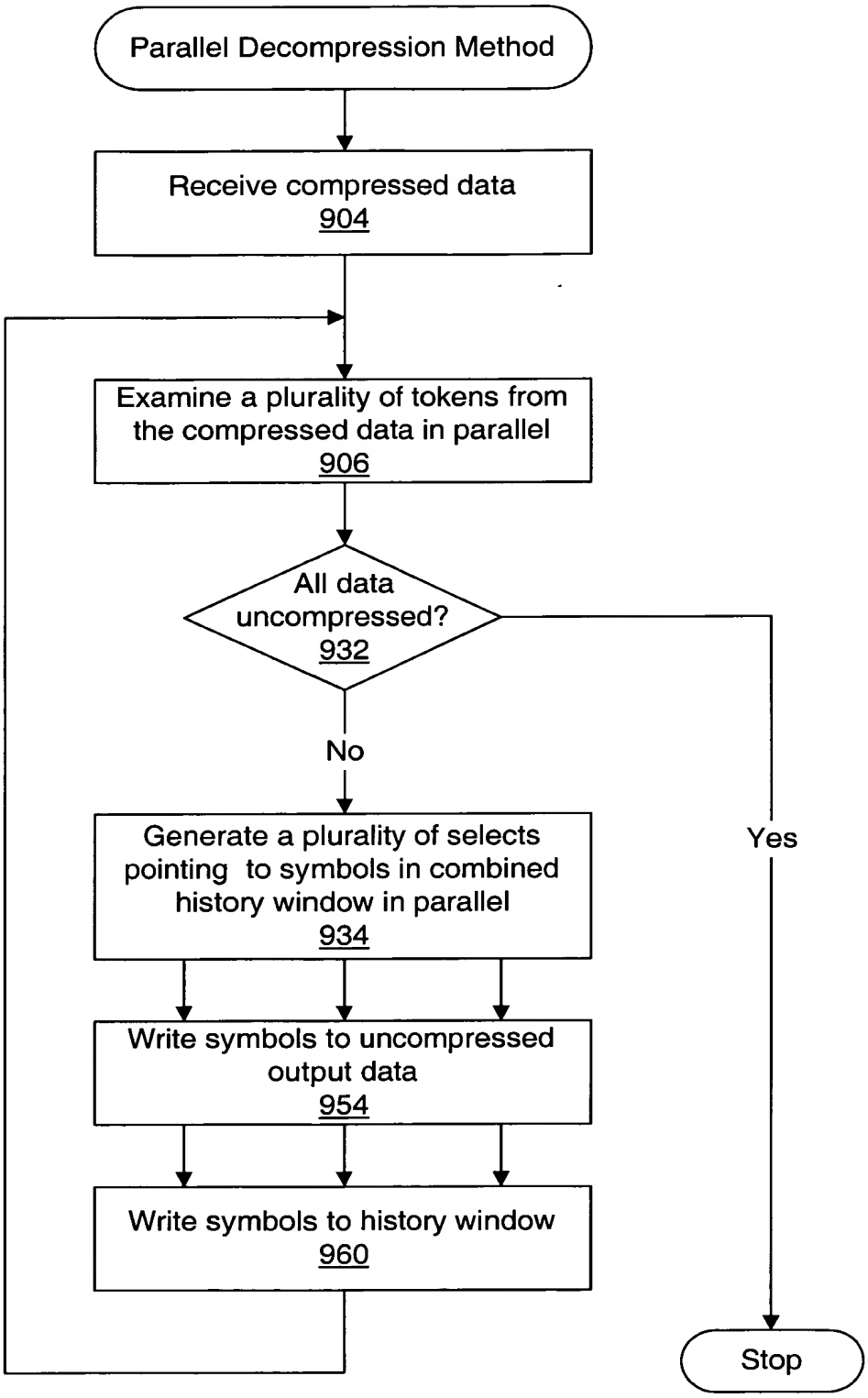


Figure 43b
Decompresion flowchart

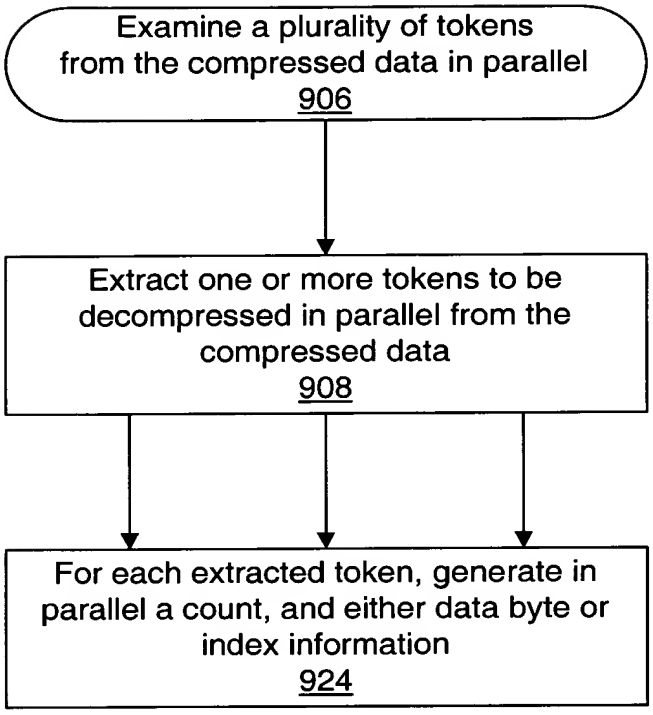


Figure 43c
Decompresion flowchart

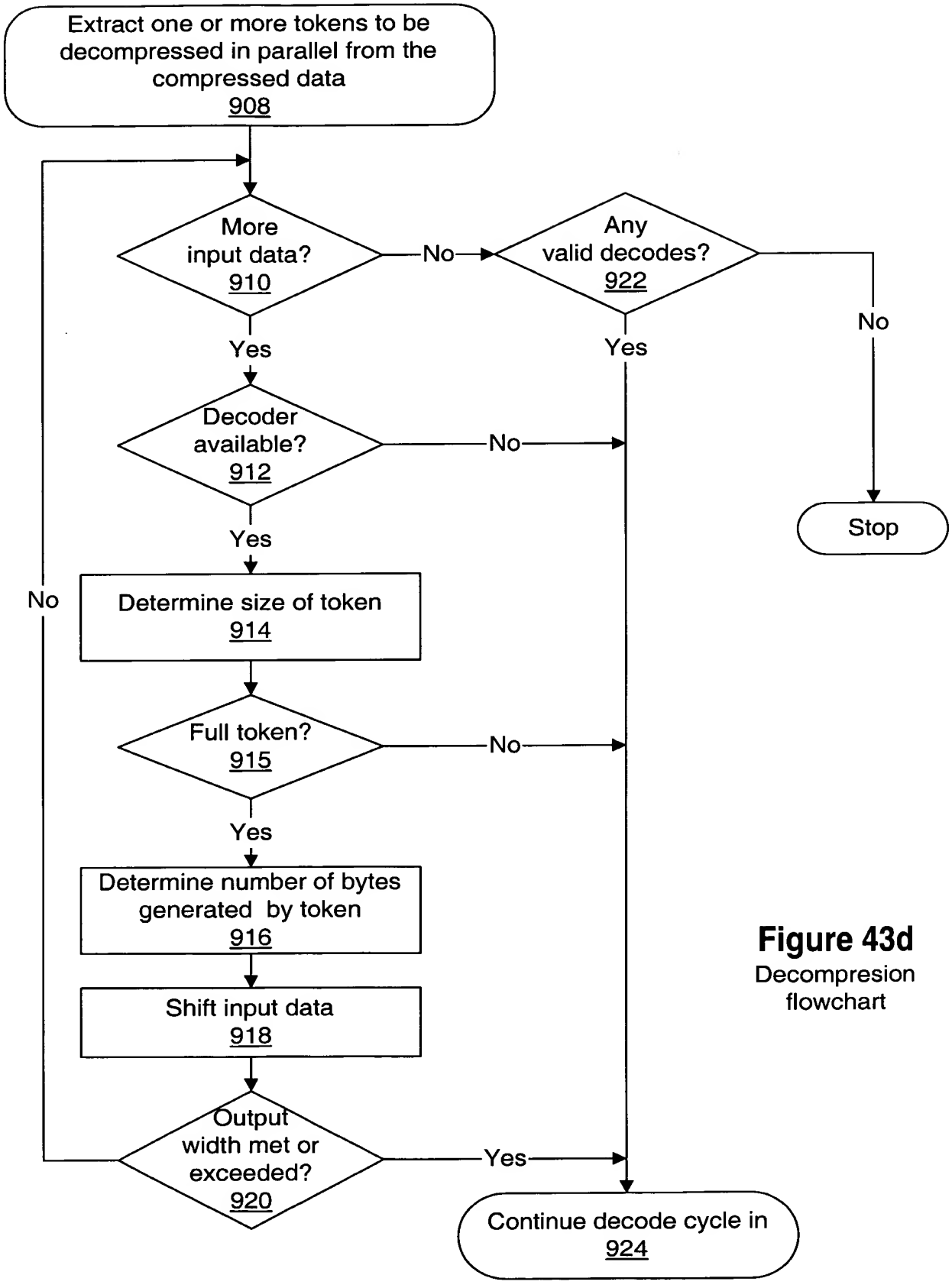


Figure 43d
Decompresion flowchart

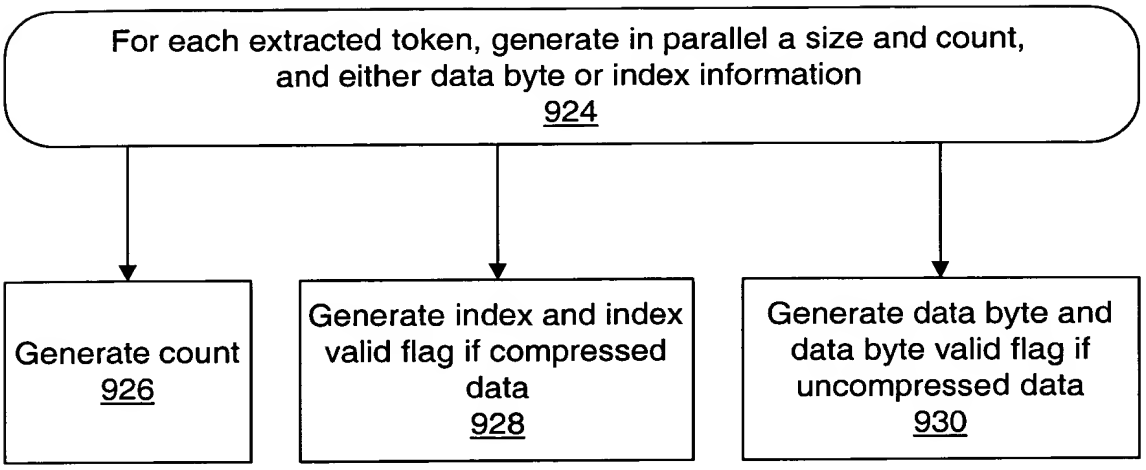


Figure 43e
Decompresion flowchart

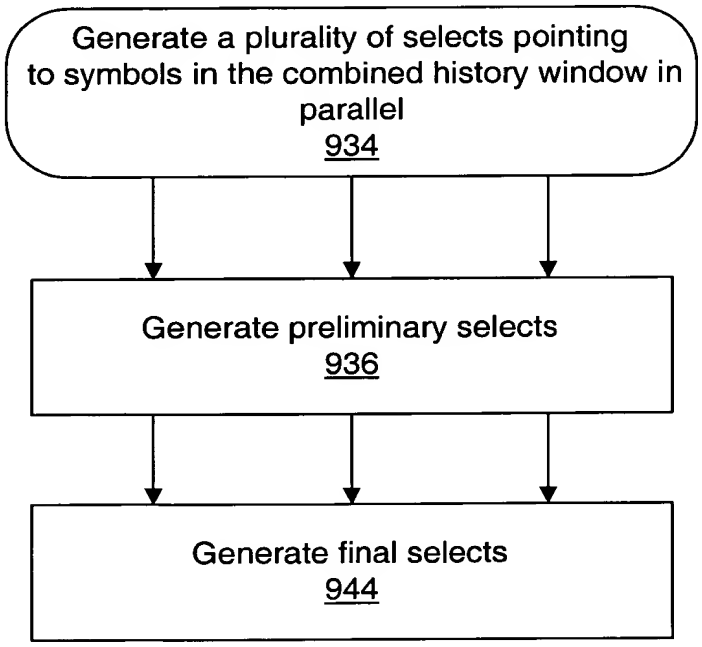


Figure 43f
Decompresion flowchart

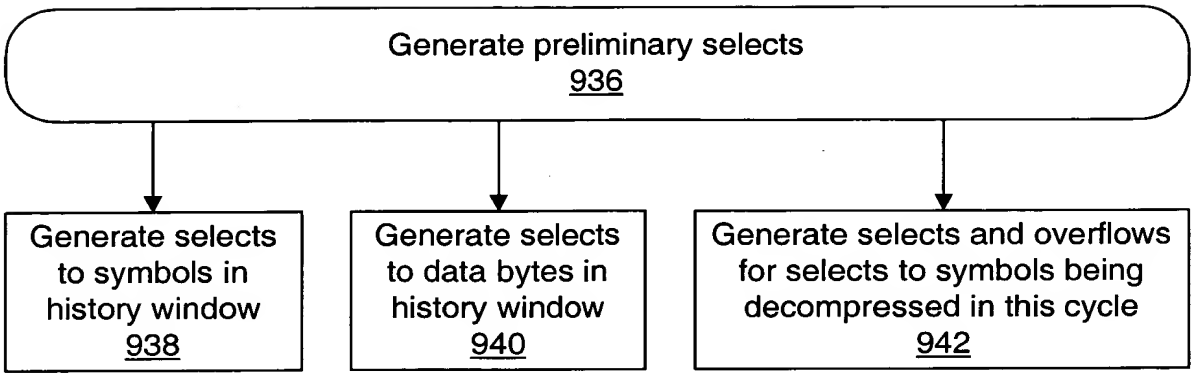


Figure 43g
Decompresion flowchart

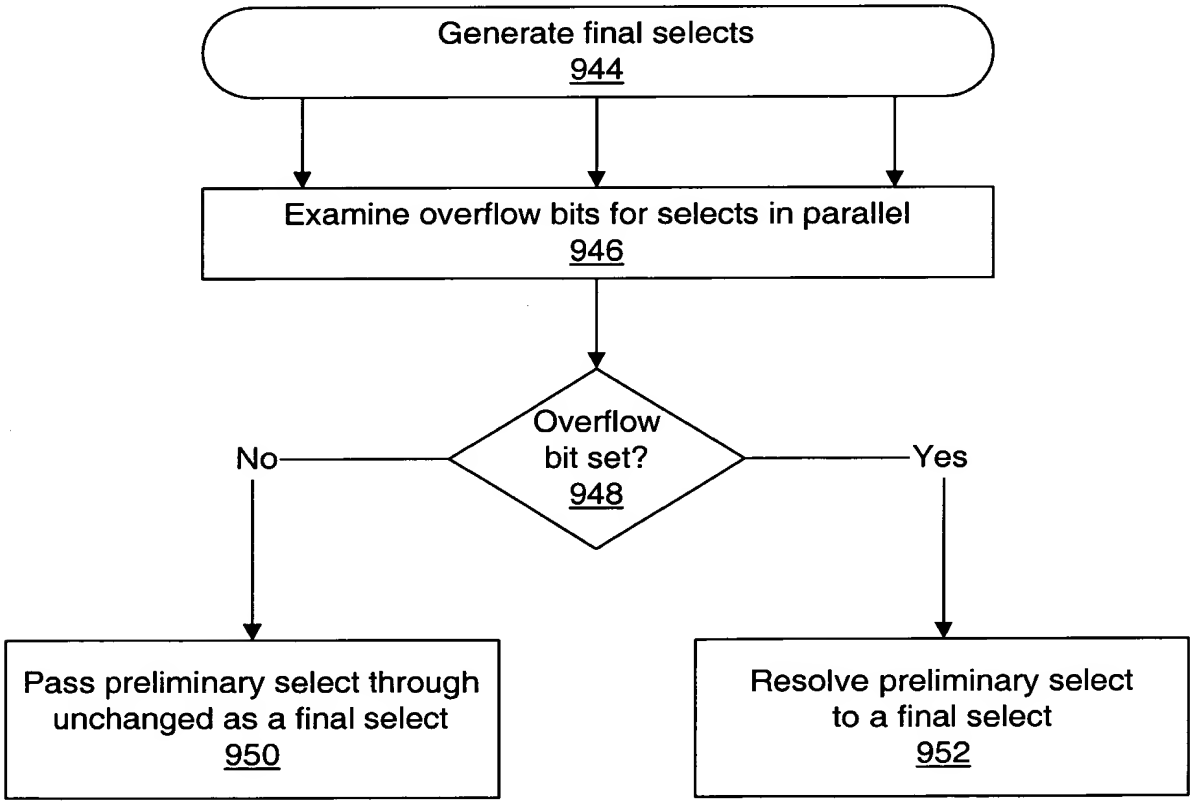


Figure 43h
Decompresion flowchart

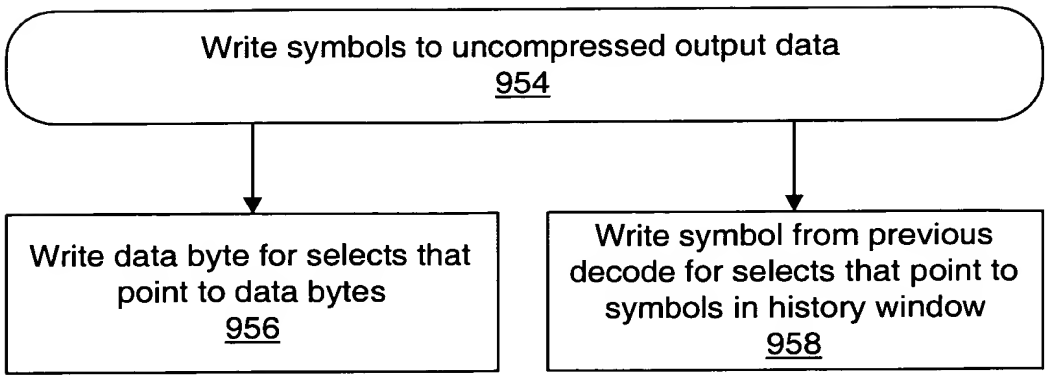


Figure 43i
Decompresion flowchart

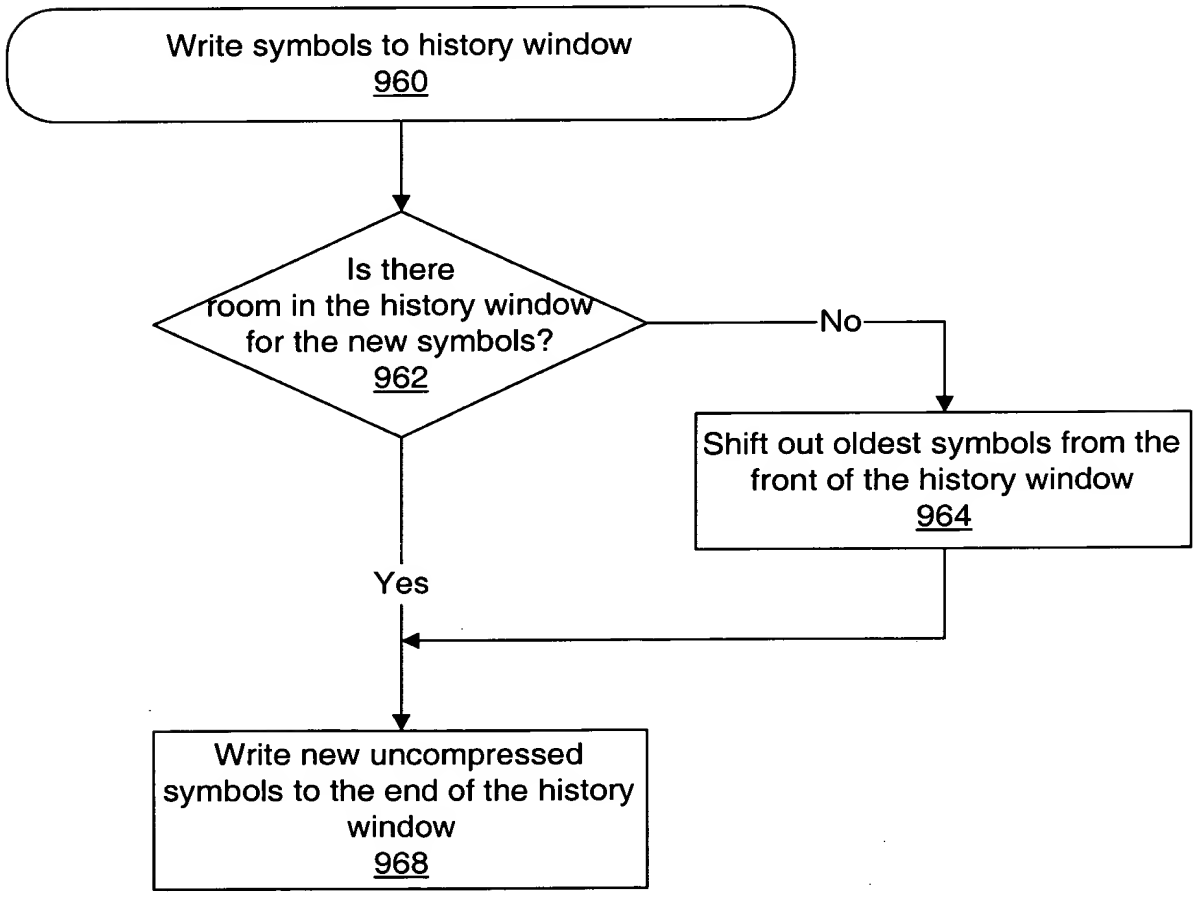


Figure 43j
Decompresion flowchart

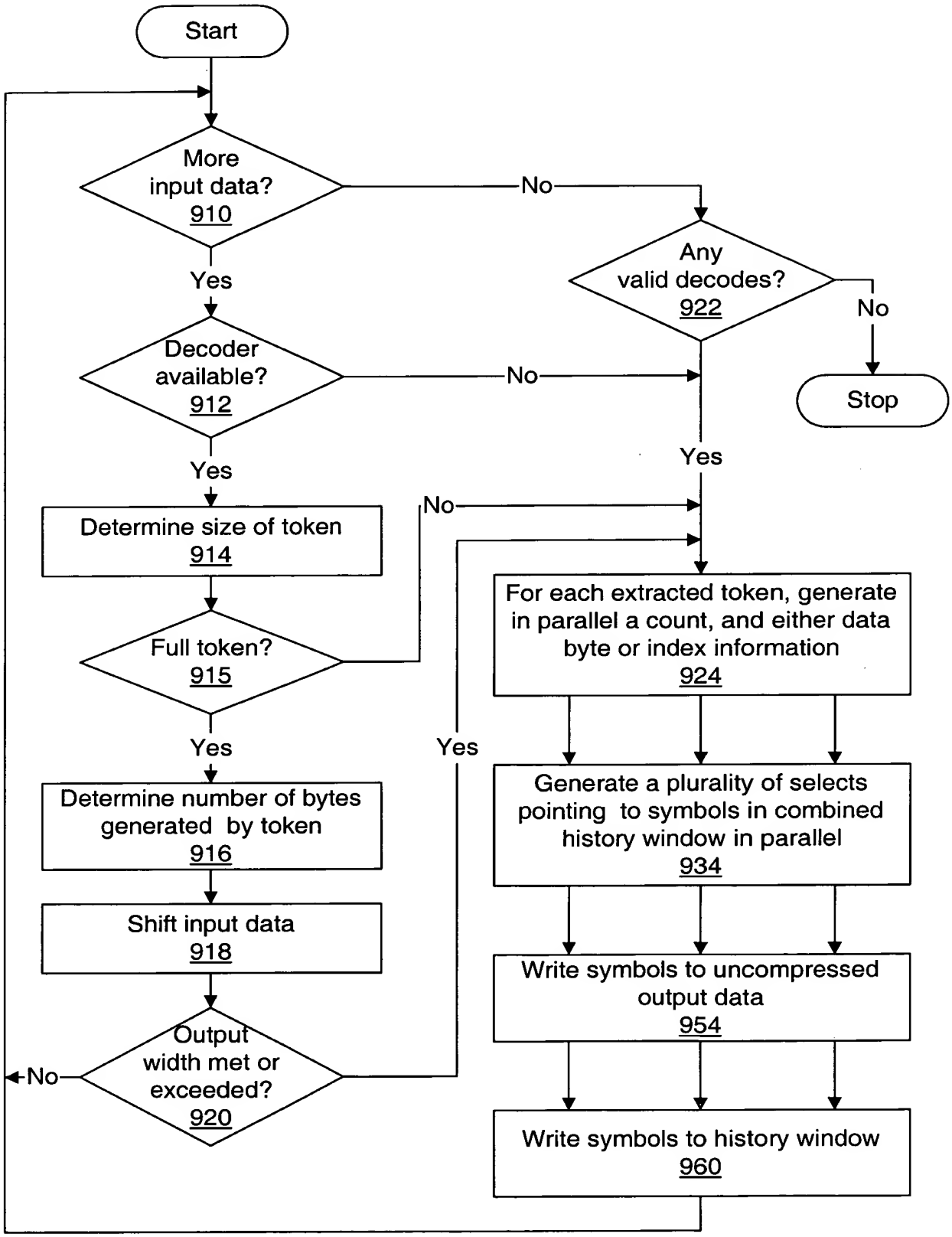


Figure 43k

Decompresion flowchart

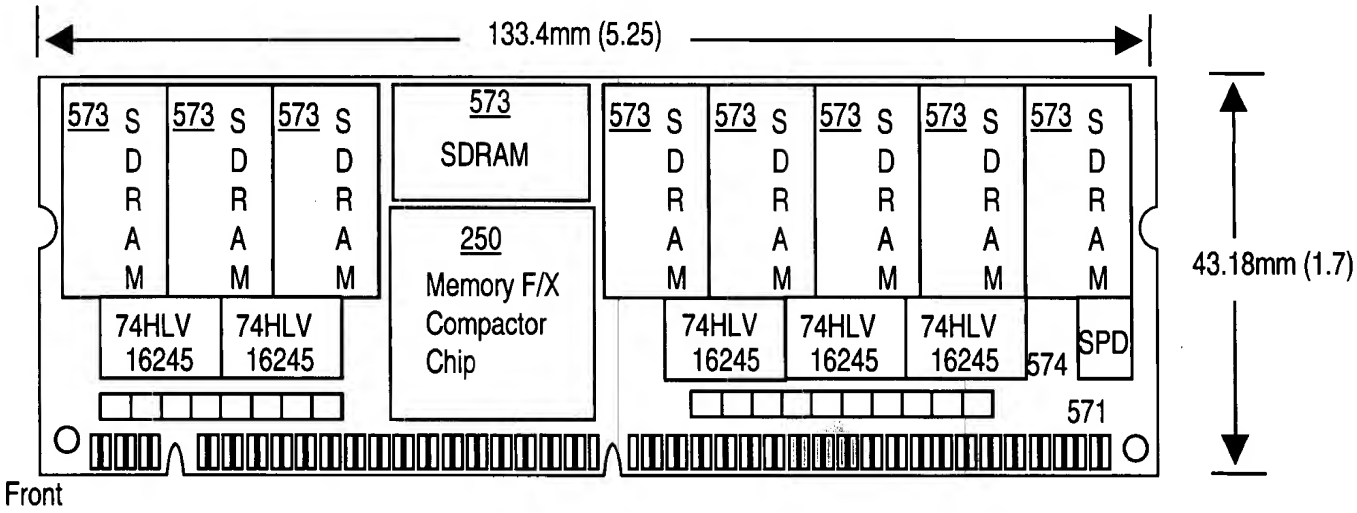


Figure 44a

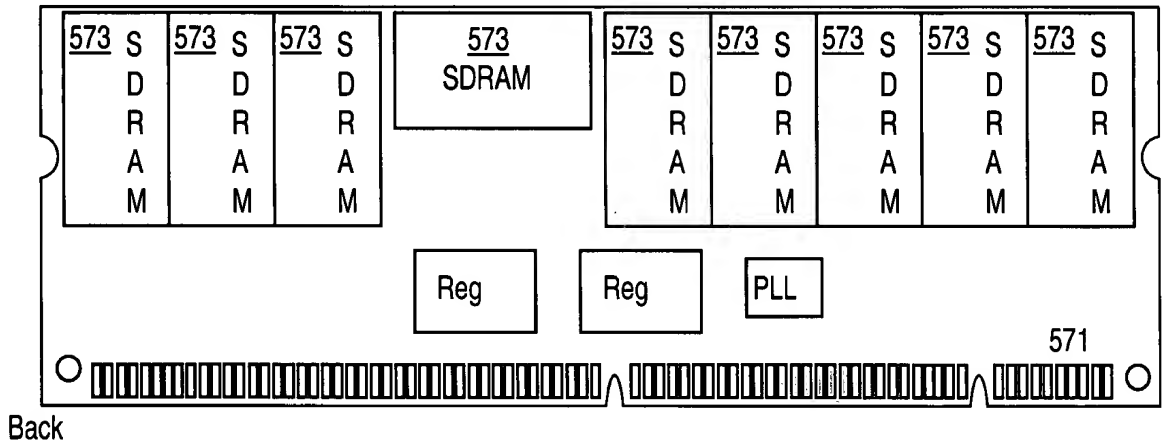


Figure 44b

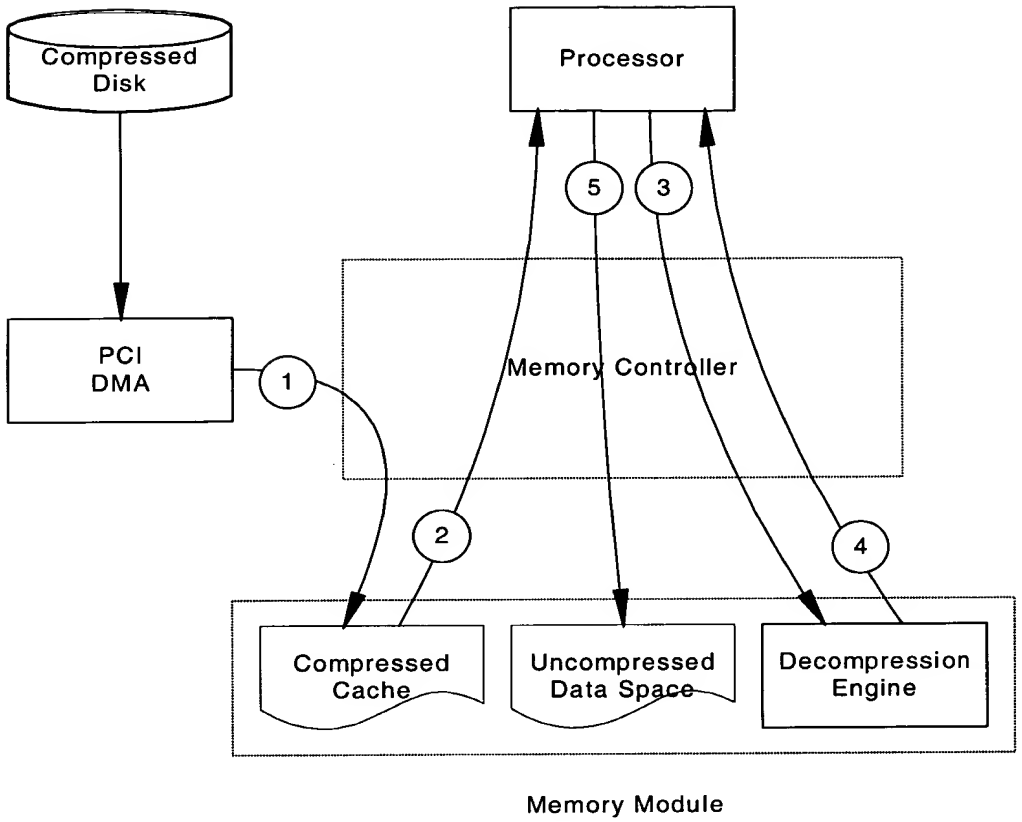

Compressed Disk
Processor
5
3
PCI DMA
1
Memory Controller
2
4
Compressed Cache
Uncompressed Data Space
Decompression Engine
Memory Module

Fig. 45

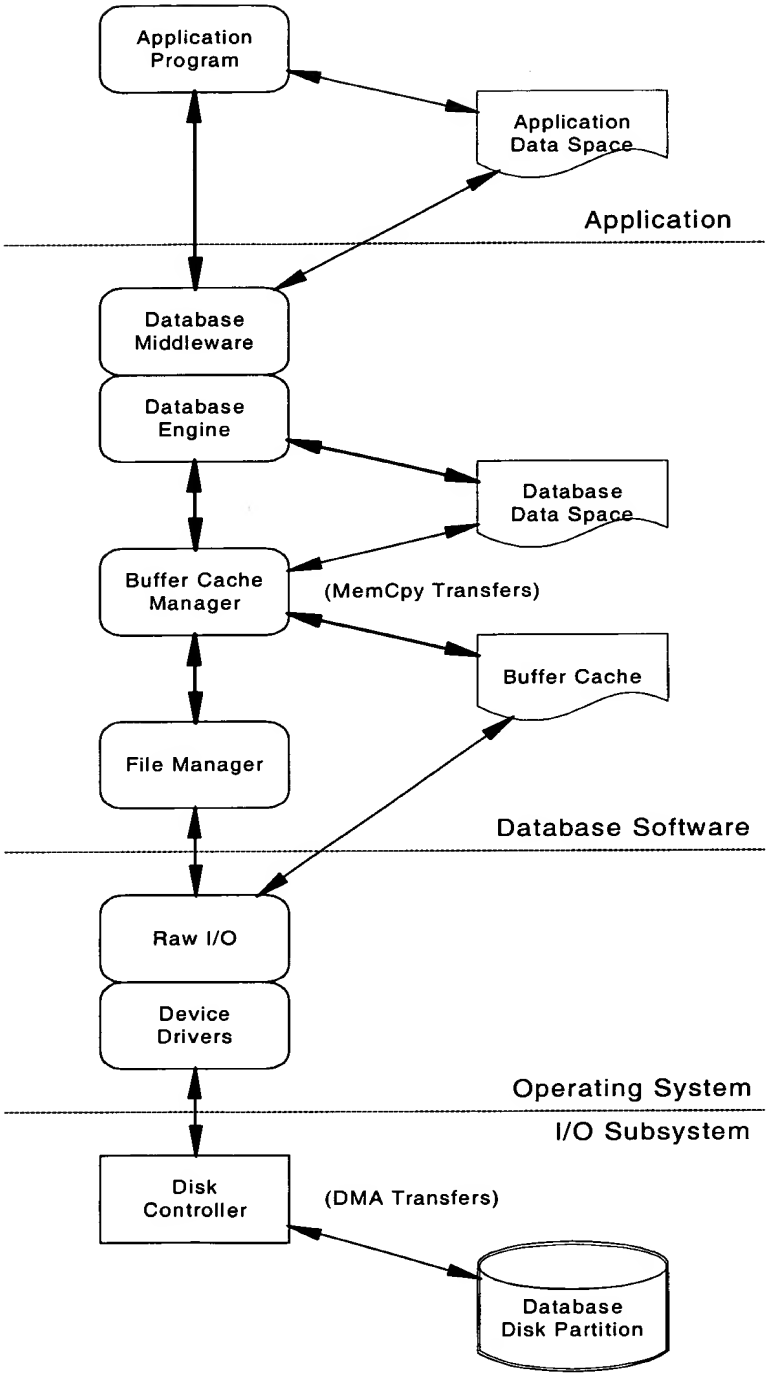


Fig. 46 Typical Database System

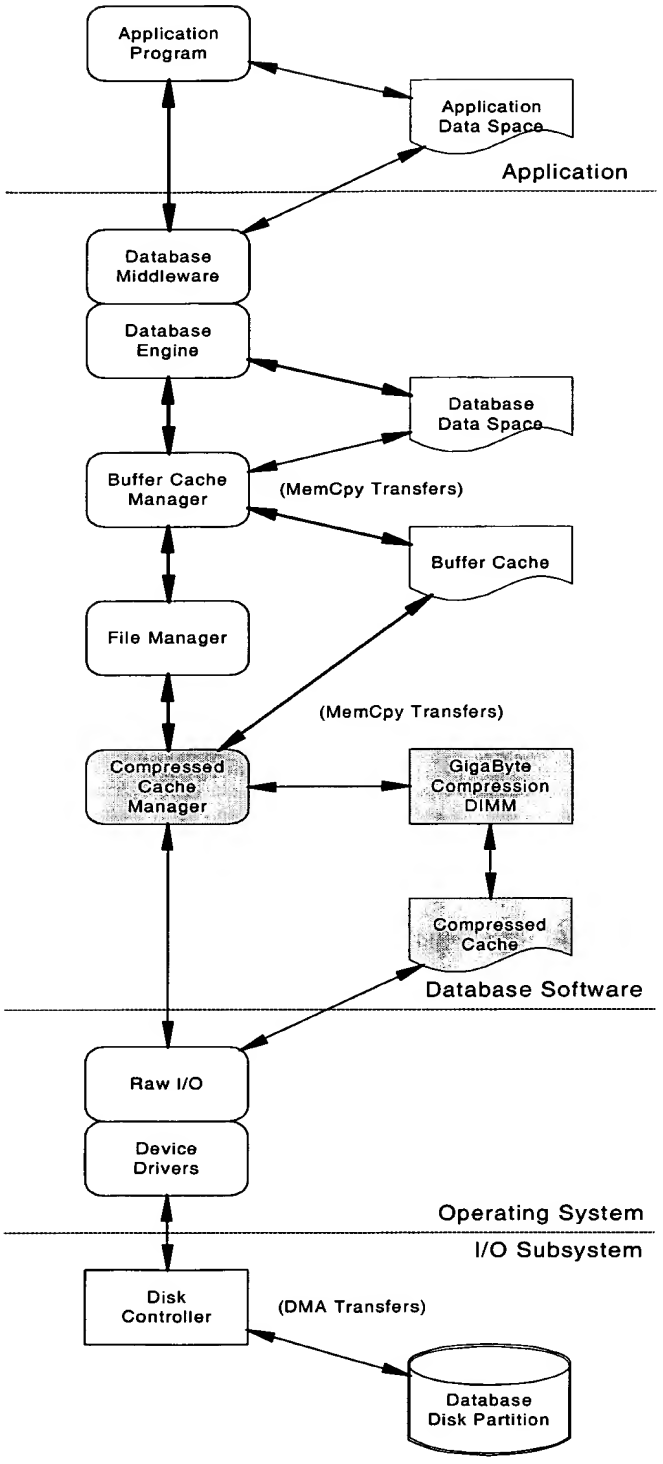


Fig 47 File System Filter Example

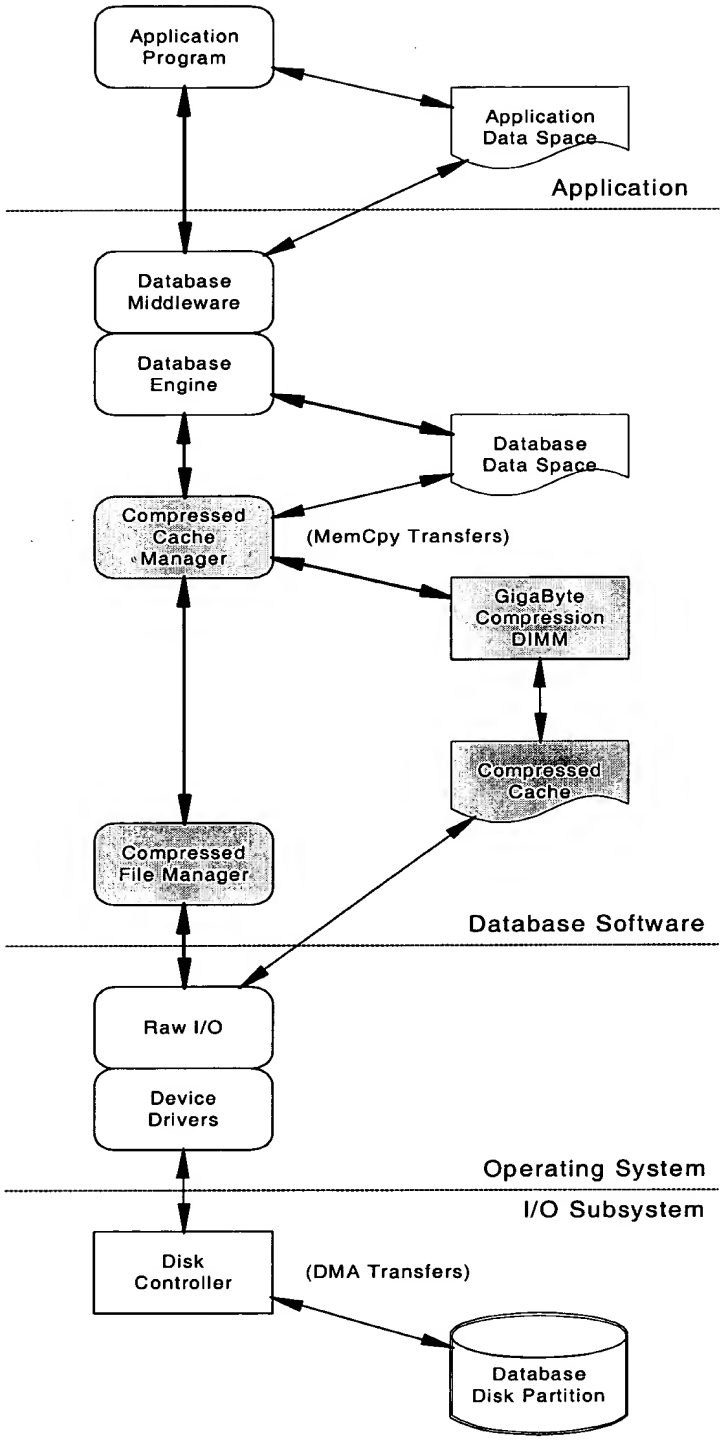


Fig. 48 Compressed Buffer Cache Example